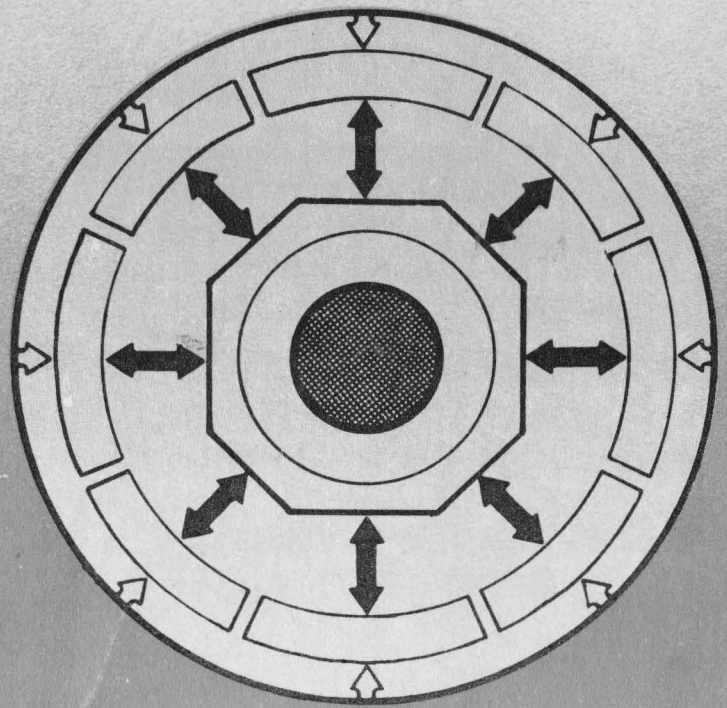
PETREA TABARCEA • GHEORGHE GHIUR

# Sisteme de inteligență artificială și roboți



# Introducere

Apariția unor mijloace tehnice noi și a tehnologiilor de mare productivitate reprezintă o caracteristică importantă a fiecărei epoci din istoria civilizației umane. Dacă inventarea mașinii cu vapori a însemnat trecerea la prima revoluție tehnică, etapă ce caracterizează tranziția de la era uneltelor la era mașinilor, iar, spre finele secolului al XIX-lea, odată cu apariția motoarelor cu combustie internă, cu extinderea energiei electrice și afirmarea industriei chimice, istoria omenirii a înregistrat o nouă revoluție tehnică, revoluția energetică, în prezent ne găsim în plină revoluție a mașinilor-creier, declanșată de automatizarea bazată pe tehnica electronică, introducerea și proliferarea echipamentelor de calcul și de prelucrare a datelor în toate domeniile de activitate, cu implicații asupra creșterii considerabile a volumului și ritmului proceselor industriale.

Analiza dezvoltării economice și sociale în multe din țările lumii atestă ponderea deosebită a ramurilor industriale noi, de înaltă tehnicitate și în mod deosebit a electronicii, microelectronicii, automaticii și tehnicii de calcul care dețin un rol determinant în afirmarea revoluției științifice și tehnice contemporane.

În cadrul politicii științifice de industrializare elaborate de Partidul Comunist Român un scop primordial îl reprezintă ridicarea nivelului tehnic și calitativ al întregii producții materiale, afirmarea cu și mai mare intensitate a revoluției științifice și tehnice contemporane, creșterea cu prioritate a producției de utilaje și instalații tehnologice, construcții navale, industriale, electrotehnică,

mașini-unelte, mecanică fină. În acest sens tovarășul Nicolae Ceaușescu, secretar general al Partidului Comunist Român, președintele Republicii Socialiste România, arăta că „*dispunem de o puternică bază umană, tehnică și materială care ne permite să ne angajăm în cercetarea celor mai complexe probleme.*

*Cercetarea științifică și tehnologică trebuie să pornească permanent de la concepția revoluționară, materialist-dialectică, de la faptul că procesul cunoașterii nu s-a încheiat și nu se va încheia niciodată, că există numai probleme necunoscute, dar care pot fi descoperite și cunoscute.*“ [1]

Pentru toate domeniile industriei noastre socialiste electronica și automatica, la care se adaugă tehnica de calcul, constituie principalele elemente generatoare de progres tehnic, contribuind decisiv la creșterea productivității muncii, la ridicarea calității produselor, la introducerea în fabricație a tehnologiilor noi, la perfecționarea metodelor de conducere și organizare a producției.

Industria noastră electronică s-a dezvoltat în mod accentuat în toate ramurile sale — electronică industrială, sisteme electronice de calcul, echipamente de testare și măsură, comunicații și componente electronice, bunuri de larg consum —, multe din realizările ultimilor ani, printre care televizoarele alb-negru și color, radioreceptoarele portabile, minicalculatoarele din familiile INDEPENDENT și CORAL, echipamentele periferice și terminale, sistemele de recepție a imaginilor meteorologice de la sateliții METEOSAT, echipamentele de transmisie a datelor, aparatura de măsură tensometrică etc., fiind cunoscute și apreciate nu numai de beneficiarii din țară, ci și de cei de peste hotare.

Consfințind rolul electronicii și informaticii în îndeplinirea deziderатului ca deceniul 1981—1990 să devină deceniul științei, tehnicii, calității și eficienței, Legea Planului național unic de dezvoltare economico-socială a Republicii Socialiste România pe perioada 1986—1990, elaborată în spiritul Directivelor Congresului al XIII-lea

[1] Nicolae Ceaușescu, *Cuvîntare la Congresul științei și învățămîntului*, Editura Politică, București, 1985, p. 9.



al partidului, precizează că industria electronică va pătrunde masiv în întreaga activitate economico-socială fiind orientată cu precădere spre dezvoltarea producției de componente electronice, mijloace de automtizare, echipamente de electronică industrială și profesională. În domeniul mijloacelor tehnicii de calcul electronice, creșterea producției se va realiza îndeosebi prin asimilarea de noi minicalculatoare pentru conducerea proceselor industriale și a unor tipuri evoluate de echipamente periferice. Pornind de la aceste sarcini deosebit de importante, *Programul de cercetare științifică, dezvoltare tehnologică și de introducere a progresului tehnic în perioada 1986—1990*, document adoptat de Congresul științei și învățămîntului, acordă o mare atenție dezvoltării în continuare a electronicii, intensificării activității de cercetare în acest domeniu. Denumită ramură de înaltă tehnicitate, generatoare de progres tehnic, industria electronică va înregistra în acest cincinal o creștere de 1,74 ori. Aplicarea largă a sistemelor tehnice de calcul, informaticii, aparaturii de măsură și control în toate ramurile economiei noastre naționale va contribui la înfăptuirea programelor de creștere mai accentuată a productivității muncii, la ridicarea calității și fiabilității produselor — prin asigurarea unui flux de fabricație cu creșterea gradului de tipizare de la 55%, în prezent, la 90%, în anul 1990 — și la reducerea consumurilor specifice de materiale cu 10—20%.

În domeniul componentelor electronice, de exemplu, cercetarea științifică va avea în atenție asigurarea producerii de circuite integrate pe scară largă, tranzistoare de putere și comutație, componente piezoelectrice, filtre cu cuarț termostabilizate, componente optoelectronice, dezvoltîndu-se în perioada 1986—1990, prin forțe proprii, peste 100 de tehnologii de nivel tehnic mondial ridicat.

În prezent progresul microelectronicii este deosebit de amplu, întrucît circuitele integrate reprezintă o nouă treaptă în evoluția electronicii, ele realizînd nu numai funcții foarte complexe, ci fiind și purtătoare de inteligență sub formă de programe de calculator executabile sau memorate. Electronica a evoluat spre așa-numita

electronică funcțională, care a intrat în noi raporturi cu omul, cu societatea.

Preocupările și eforturile constante ale unor cercuri largi de specialiști români sînt orientate tocmai în această direcție, pentru proiectarea și realizarea unor produse de nivel tehnic ridicat care să răspundă cerințelor tot mai diverse ale utilizatorilor din toate domeniile activității economice și sociale, de la industrie și agricultură, construcții și transporturi, pînă la medicină, educație, știință și cultură.

Conducerea proceselor industriale cu calculatoare electronice din familia FELIX, minicalculatoare și microcalculatoare performante — I 102 F, M 18, CORAL 4030 — reprezintă una din principalele utilizări ale sistemelor electronice de calcul produse în țara noastră, cu aplicații în întreprinderi și institute de cercetare din metalurgie, siderurgie, construcții de mașini, chimie, industria ușoară și alimentară etc., determinînd, alături de alți factori, creșterea gradului de organizare și a eficienței producției.

Proiectarea asistată de calculator a devenit un mijloc avansat de realizare a unor economii importante de materii prime și materiale, energie și combustibil încă din fazele de gîndire și concepție a viitoarelor produse, de simplificare a muncii omului pentru îndeplinirea obiectivelor complexe referitoare la dezvoltarea multilaterală a țării. Sistemele pentru proiectarea asistată de calculator a circuitelor imprimate, a SDV-urilor, precum și cele determinante pentru atingerea calității impuse produselor destinate pieței interne și exportului sînt exemple semnificative în acest sens.

Alături de alte domenii, medicina se bucură de atenția specialiștilor români în electronică și tehnică de calcul, pentru acest sector de activitate fiind în curs de finalizare echipamente de analiză și monitorizare capabile să proceseze în timp real o mare cantitate de informații în vederea luării unor decizii rapide și exacte în lupta cu secundele pentru salvarea vieții omului.

Alte sisteme de cel mai înalt nivel tehnic sînt destinate supravegherii traficului rutier, optimizării consumului de carburant de către autoturisme, transmisiilor de



date, comunicațiilor și telecomunicațiilor inclusiv prin sateliți, recepționării informațiilor de la sateliți etc., în aplicații care permit extinderea permanentă a limitelor cunoașterii umane.

Diversificarea accentuată a produselor electronice, bazată pe microminiaturizare și integrare, robotizare și introducerea inteligenței artificiale și a sistemelor-expert conduce la un impact puternic asupra tuturor domeniilor vieții economice și sociale, care transformă radical viața noastră de zi cu zi. O simplă enumerare a unora dintre aplicațiile cele mai noi ale electronicii, tehnicii de calcul și microelectronicii deschide o perspectivă nebănuită chiar și specialiștilor în materie : sisteme de rețele de calculatoare electronice interconectate ; accesul direct al persoanelor la rețele de date prin linii telefonice obișnuite ; sisteme interactive de informații la domiciliu folosind televizorul ; traducerea automată a textelor ; diagnosticarea cu ajutorul calculatorului, în medicină ; consultații medicale la domiciliu prin calculator electronic ; sisteme electronice pentru educație ; administrația fără hîrtie, prin utilizarea largă a calculatoarelor ; poșta electronică etc.

Dezvoltarea electronicii și tehnicii de calcul, a componentelor electronice și roboticii va influența toate ramurile economiei naționale, va reprezenta suportul transformării considerabile a structurilor industriale ale acestora, în conformitate cu cerințele noii revoluții științifice și tehnice. Tehnica de calcul, semiconductoarele, roboții industriali și echipamentele de telecomunicații sînt considerate pretutindeni factori hotărîtori ai dezvoltării economice, determinînd ridicarea continuă a performanțelor produselor, creșterea productivității muncii, reducerea consumurilor energetice și de materii prime.

În acest context, lucrarea — structurată pe patru capitole — cuprinde un spectru larg de aspecte puțin evidențiate sau insuficient tratate în literatura de specialitate din țara noastră privind problematica inteligenței artificiale, a sistemelor-expert și a roboților.

În prima parte a lucrării sînt analizate cele mai semnificative aspecte referitoare la inteligența artificială, la raportul ei cu inteligența naturală a omului, insistîn-

du-se în special asupra tehnologiei inteligenței artificiale și programelor de inteligență artificială cu utilizările lor. Întrucît evoluția și revoluția în informatică depind de limbajele de programare și de complexitatea programelor informatice sînt ilustrate principalele caracteristici ale programelor de inteligență artificială, a căror apariție are semnificația unui salt revoluționar. În a doua secțiune a lucrării atenția este îndreptată spre studiul sistemelor-expert și al roboților, relevîndu-se multitudinea aplicațiilor acestora. Din examinarea generațiilor de roboți, a raportului acestor dispozitive complexe cu omul se poate desprinde concluzia potrivit căreia asemenea tehnologii și utilaje permit constituirea unei științe a producției în care formalizarea și implicit automatizarea proceselor de producție să poată fi cît mai avansate.

Chiar dacă spațiul nu a permis o abordare exhaustivă a fiecărui aspect circumscris domeniilor menționate, bibliografia va conduce cititorul interesat către acele lucrări în care elementele prezentate în această carte sînt tratate în detaliu.

Elaborînd acest volum autorii își exprimă convingerea că el va fi util unui număr mare de cititori, electroniști, automaticieni, specialiști în tehnica de calcul, informaticieni, studenți din cadrul institutelor de profil sau elevi, tuturor celor implicați, pasionați sau numai interesați în dezvoltarea electronicii, informaticii și automatizării — domenii de maximă importanță pentru progresul tehnico-științific.

# 1

# Inteligența artificială

## 1.1. CONCEPTUL DE INTELIGENȚĂ ARTIFICIALĂ. INTELIGENȚA ARTIFICIALĂ ȘI INTELIGENȚA NATURALĂ. INGINERIA CUNOAȘTERII

Preocuparea specialiștilor de a crea programe pentru calculatoare electronice „inteligente“ — sisteme care prezintă caracteristici asociate cu inteligența umană (înțelegerea vorbirii, învățarea, judecata și rezolvarea problemelor) —, element la care cu siguranță putem însuma salturile produse în electronică și informatică, în domeniul sistemelor electronice de calcul sub impactul noii revoluții științifice și tehnice a condus la apariția unui domeniu interdisciplinar, cu aplicații practice nebănuite sub denumirea generică de *inteligență artificială*.

Mai limitată pentru început, inteligența artificială (AI — *Artificial Intelligence)* a devenit o realitate, bazîndu-se pe posibilitatea pe care o au calculatoarele electronice de a admite programe cu caracter de inteligență, de a prelucra date în conformitate cu aceste programe. Inteligența artificială se reduce la anumite tipuri de programe de prelucrare a simbolurilor, ceea ce ne arată că AI este o proprietate a unor structuri informaționale dinamice, care, în principiu, se pot desfășura pe o structură de neuroni sau într-un set de module nervoase, într-un sistem electronic de calcul ori pe o plachetă de siliciu atunci cînd calculatorul se reduce la o singură structură.

Există numeroase definiții ale conceptului de inteligență artificială. N. Findler, cercetător cunoscut în acest domeniu, arată că „un sistem este considerat a avea pro-

prietatea de inteligență, pe baza observării comportării sistemului, dacă se poate adapta singur la noi situații, are capacitatea de a raționa, de a înțelege legăturile dintre fapte, de a descoperi înțelesuri și de a recunoaște adevărul. De asemenea, ne așteptăm ca un sistem inteligent să învețe ; cu alte cuvinte, să-și îmbunătățească nivelul performanțelor pe baza experienței trecute" [1]. După cum remarcă Mihai Drăgănescu, în definiția de mai sus nu se spune nimic despre creație. „Pentru inteligență, în sensul definiției amintite, lumea este dată, inteligenței nu-i rămîne decît s-o cunoască, să determine adevărul și să acționeze în conformitate cu cunoașterea pe care o obține" [2].

Un alt cercetător al conceptului, Patrick Winston, apreciază că inteligența are anumite atribute — proprietățile de a raționa, de a asimila, obține și aplica cunoaștere, de a manipula și comunica idei ; o definiție este totuși dificilă, întrucît „inteligența apare a fi un amalgam cu foarte multe proprietăți de reprezentare și prelucrare a informației" [3].

În sens restrîns, AI este un program de calcul [4]. Această afirmație se bazează pe faptul că atît programele, cît și calculatoarele electronice au proprietatea de a reprezenta situațiile cele mai diferite prin simboluri și operații logice.

Este important de a releva deosebirea existentă între inteligența artificială și inteligența naturală, deosebire pe care știința ar putea s-o confirme sau s-o infirme în viitor. Și în acest domeniu părerile specialiștilor sînt împărțite. De exemplu, Patrick Winston, care consideră că inteligența artificială este identică cu cea naturală se întreabă „Ce poate să însemne cînd spunem că un calculator a înțeles ceva ? Este oare suficient că acest calculator ne oferă un răspuns inteligent ?" [5]. Desigur, nu este suficient, întrucît atît la om, ca și la animal, se observă că, spre deosebire de tot ceea ce aparține domeniului inteligenței artificiale, mai apare fenomenul de conștiență — o realitate obiectivă. Așadar, inteligența naturală este inteligența artificială la care se adaugă un element care semnifică influența conștienței asupra inteligenței omului.

[1] Findler, N. V. *Artificial Intelligence*, în *Encyclopedia of Computer Science*, New York, Petrocelli/Charter, 1976, p. 106—116.

[2] Drăgănescu, M., *A doua revoluție industrială. Microelectronica, automatica, informatica — factori determinanți*, Editura Tehnică, București, 1980, p. 20.

[3] Winston, P., *Artificial Intelligence*, Addison-Wesley Publishing Co., Reading Massachussets, 1977, p. 1.

[4] Drăgănescu, M., *Op. cit.*, p. 22.



O problemă importantă în domeniul AI este cea a limbajului [6]. Un program de AI înțelege realitatea, pentru un domeniu limitat al realității. Între acțiuni și limbaj există o strînsă legătură, dar în spatele acestor două noțiuni se află un program și numeroase simboluri. Programul este cel care vorbește corect din punct de vedere gramatical într-un univers restrîns ce impune foarte multe constrîngeri limbajului. În consecință, semantica [7] este mult simplificată.

În domeniul AI se încearcă, în ceea ce privește limbajul, introducerea unor structuri mai profunde cu ajutorul reprezentării prin cadre *(frames)* [8]. Aceste frame — „un fel de fișe" [9] — cuprind cuvinte, dar în locul cuvintelor putem avea orice simboluri sau orice stări. Cuvintele din frame corespund unor noțiuni sau unor ac-

[5] Winston, P., *Op. cit.*, p. 22.

[6] Mijloc de transmitere a informației între membrii unei categorii de indivizi, caracterizat, în general, prin formele sale de reprezentare a informației, împreună cu regulile de organizare a acestor forme, prin „înțelesul", semnificația asociată formelor respective, împreună cu regulile de constituire a înțelesului unor forme de reprezentare pe baza „înțelesurilor" componentelor și, eventual, prin modul în care limbajul urmează a fi folosit de către diferiți indivizi (v. *Dicționar de informatică*, Editura Științifică și Enciclopedică, București, 1981, p. 192).

[7] Set de reguli ce determină „înțelesul", semnificația propozițiilor într-un limbaj. Este vorba, în esență, de reguli de evaluare a acestor propoziții în termenii unor mulțimi de valori bine cunoscute celor ce folosesc limbajul respectiv.

[8] Zone transversale din banda magnetică pe care se poate înregistra un caracter (7—9 biți). În unele lucrări de specialitate (v. Drăgănescu, M., *Op. cit.*, p. 24) în locul termenului de *cadre* este folosit cel de *frame*.

[9] Drăgănescu, M., *Op. cit*, p. 24.

țiuni, legăturile dintre acestea constituind un sistem de frame care reflectă o cunoaștere [10] dintr-un punct de vedere. În domeniul AI problema limbajului natural nu este încă soluționată. De aceea mulți specialiști apreciază că clarificarea limbajului natural reprezintă problema centrală a inteligenței artificiale.

Concluzionînd, „inteligența artificială pare a fi o proprietate a structurilor informaționale dinamice, construite din simboluri, indiferent de forma și suportul lor fizic, implicînd ordonarea sintactică a simbolurilor și o semantică de ordinul I, caracterizată prin recepționarea de informații externe ei, pe plan pur informațional sau în raport cu o realitate materială (substanțială), prin construirea de modele proprii și cunoaștere asupra acesteia, inclusiv de comunicare către aceasta" [11].

În ultimii 30 de ani evoluția tehnicilor de inteligență artificială a impus abordarea unei tematici variate pentru rezolvarea unor probleme legate de jocurile de șah sau dame pînă la sistemele-expert utilizate în toate ramurile vieții economice și sociale. În acest sens s-au dezvoltat noi modalități de lucru prin investigarea rețelei sau structurii arborescente a stărilor consecutive generate de sisteme.

Principiile stabilite pentru programele inteligente au dat naștere ingineriei cunoașterii, care deschide o nouă perspectivă capacității creatoare a minții omului. În tabelul 1.1 este ilustrată o comparație sugestivă între tehnica prelucrării datelor folosită în prezent și ingineria cunoașterii.

În țara noastră preocupările pentru dezvoltarea activităților legate de inteligența artificială au permis formarea unor colective puternice — în principal în cercetare și învățămînt — care au reușit să concretizeze în

[10] Categorie filozofică tratînd relațiile dintre subiect și obiect, modelul general, de principiu, al asimilării și reconstrucției informaționale a lumii obiective de către subiect.

[11] Drăgănescu, M., *Op. cit.*, p. 25.



TABELUL 1.1. **Comparație între prelucrarea datelor și ingineria cunoașterii**

| Caracteristici | Prelucrarea datelor | Ingineria cunoașterii |
|---|---|---|
| **Entități** specifice | Date | Cunoștințe |
| **Modul** de reprezentare | Algoritmic | Euristic |
| **Tipul** procesului | Cicluri repetitive | Inferențe |
| **Manipularea** efectivă | Baze de date | Baze de cunoștințe |

aplicații practice — limbaje noi sau extinse și sisteme-expert — lucrările de pionierat ale unor specialiști români.

## 1.2. EVOLUȚIA INTELIGENȚEI ARTIFICIALE

Inteligența artificială s-a constituit în ultimele trei decenii ca un domeniu interdisciplinar cu aplicații practice importante în care își aduc contribuția specialiști cu profesii și preocupări diferite — electroniști, automaticieni, ciberneticieni, matematicieni, psihologi, lingviști, filozofi etc. În alte sfere de activitate — medicină, chimie, biologie, geologie etc. — inteligența artificială se impune sub forma sistemelor-expert pentru stabilirea diagnosticelor, interpretarea datelor, prelucrarea simbolică etc. prin tehnici de deducție-inferență și modele simbolice.

Inteligența artificială s-a născut din dorința de a elabora un program de șah care să joace foarte bine. Dar forța motrice a acestei discipline își are originea în ambiții cu mult mai mari — realizarea unei inteligențe de uz general, a unei minți artificiale. La începutul anilor '50 speranțele specialiștilor în rezolvarea problemei menționate au crescut odată cu apariția și proliferarea calculatoarelor electronice.

Realizarea AI se baza pe conceptul de program — un set ordonat de proceduri — folosit de Alan Turing, John von Neumann, Norbert Wiener și Claude Shannon pentru a descrie mecanisme care să poată permite mașinilor să facă primii pași pentru a juca șah sau să-și manifeste „inteligența" în alt mod.

Pentru evoluția inteligenței artificiale se consideră că au jucat un rol deosebit logica matematică și ideile dezvoltate în anii '30 și '40 despre calculatoare. Sistemele logice prezentate în perioada menționată au demonstrat că unele aspecte ale gîndirii pot fi formalizate într-un cadru foarte simplu ; logica matematică a continuat să se afle la baza investigării inteligenței artificiale, avîndu-se în vedere că sistemele logico-deductive au fost implementate cu succes în structura calculatoarelor electronice.

Datorită lui Church, Turing și alții, ideile despre tehnica de calcul au stabilit legătura dintre formalismul gîndirii și calculatorul electronic. Un rol esențial l-a avut, în acest cadru, conceptul de prelucrare simbolică. Dar, chiar înainte de inventarea calculatoarelor digitale, Church și Turing au constatat că numerele nu reprezintă totul pentru operațiile matematice, fiind doar un mod de interpretare a stărilor interne ale mașinii. Turing, considerat întemeietorul inteligenței artificiale, a argumentat — pe baza modelului său simplu și universal de evaluare a calculelor — că mecanismul folosit poate fi înțeles ca avînd o comportare inteligentă.

Nu s-a putut fixa cu precizie o dată începînd cu care putem vorbi efectiv de AI, dar mulți specialiști utilizează ca punct de referință vara anului 1956, cînd a avut loc o conferință la Dortmouth College. Pînă atunci au existat lucrări teoretice dar scrise de oameni care au creat relativ izolat. La Dortmouth au venit oameni care au făcut practic primii pași pentru transformarea ideilor în atuuri puternice în favoarea mașinii inteligente.

Termenul de inteligență artificială a fost inventat de John McCarthy, de la colegiul amintit, într-o perioadă cînd existau puțini cercetători care investigau căile prin care mașinile ar fi putut fi astfel realizate încît să reproducă procese similare celor din creierul uman. Conferința din 1956 a reprezentat începutul inteligenței artificale ca domeniu separat al tehnicii de calcul. Propunerea lui McCarthy de organizare a unei întîlniri între puținii cercetători interesați în domeniu avea menirea de a deschide căi de comunicare între aceștia. Printre surprizele conferinței s-a numărat și prezentarea unor lucrări ale lui Allen Newell și Herbert A. Simon care au descris primul program, *Logic Theorist*, pentru calculatoare utilizate ca procesoare simbolice, de fapt primul program de inteligență artificială. Împreună cu J. C. Shaw, cei doi au creat limbajul de prelucrare a informațiilor, IPL (*I*nformation *P*rocessing *L*anguage), capabil să modeleze posibilitățile umane simple de rezolvare a problemelor.



Printre invitați s-a numărat și Marvin Minsky de la MIT (*M*assachusetts *I*nstitute of *T*echnology) — Cambridge, care lucrase anterior cu Claude Shannon de la Bell Laboratories, preocupîndu-se în mod deosebit de programarea euristică. McCarthy este creatorul limbajului de programare LISP (*LIS*t *P*rocessing) care și în prezent este unul din instrumentele de lucru favorite ale cercetătorilor.

În continuare lucrurile s-au precipitat, iar numărul cercetătorilor în domeniul inteligenței artificiale a crescut continuu. La mijlocul anilor '60 existau laboratoare specializate, dintre care amintim pe cel de la Carnegie Mellon — condus de Alan Newell și Herbert Simon —, MIT — condus de Marvin Minsky și Seymour Papert — și Stanford — condus de John McCarthy.

Încercările întreprinse pentru realizarea unei programări mai directe au condus la definirea, în anul 1970, a limbajului PROLOG (*PRO*gramming in *LOG*ic) de către Alain Colmerauer și un grup de cercetători de la Universitatea din Marsilia (Franța). Mai tîrziu, David Warren și Fernando Pereira de la Universitatea din Edinburgh (Scoția) l-au dezvoltat, limbajul impunîndu-se practic în Europa. Un salt spectaculos s-a produs abia cînd s-a aflat că proiectul de realizare a calculatorului din generația a cincea — anunțat de firme din Japonia — se bazează pe PROLOG. Acest limbaj a dat o nouă formă metodelor

mai vechi de reprezentare a cunoștințelor — logica formală. Programatorii au creat declarații sau propoziții, de fapt declarații formale asupra unor fapte logice și relații între obiecte. La fel cum LISP tratează datele și procedurile executabile ca funcții, PROLOG consideră datele și cunoștințele ca relații.

Între limbajele PROLOG și LISP există deosebiri datorate modului diferit de programare — logică și, respectiv, funcțională — astfel încît primul poate fi considerat o extensie a celui de-al doilea.

La începutul anilor '70 inteligența artificială avea toate caracteristicile unui domeniu deja bine fundamentat — congrese internaționale, reviste, cărți și cursuri la majoritatea universităților etc. În această perioadă la Stanford, sub coordonarea lui Edward Feigenbaum, a fost dezvoltat primul sistem-expert — numit DENDRAL — pentru analize chimice, stabilindu-se o direcție clară pentru evoluția sistemelor de inteligență artificială.

Există multe definiții pentru inteligența artificială, deși nici una nu a fost acceptată unanim ; între formele abstracte ca „prelucrarea informațiilor simbolice cum ar fi conceptele, cunoștințele și relațiile" și cele deterministe — „studiul tehnicilor pentru rezolvarea problemelor exponențiale în timp polinomial" —, mai completă și mai ușor de înțeles poate fi considerată următoarea definiție : inteligența artificială este acea parte a tehnicii de calcul implicată în proiectarea sistemelor de calculatoare inteligente, adică a sistemelor care prezintă caracteristici asociate cu inteligența umană — înțelegerea vorbirii, învățarea, judecata și rezolvarea problemelor.

Istoria inteligenței artificiale a stat multă vreme sub semnul cercetărilor lipsite de aplicații practice deosebite. Această stare de lucruri s-a schimbat în ultimii ani, deoarece, pe de o parte, există probleme importante de rezolvat care sînt prea complexe pentru tehnica de calcul convențională, iar, pe de altă parte, faptul că se pot obține puteri mărite de calcul la prețuri din ce în ce mai reduse a deschis cîmp larg de acțiune sistemelor practice.

Scopul declarat al cercetătorilor este realizarea sistemelor de calculatoare care să depășească posibilitățile gîndirii. Deși cercetările continuă, în prezent s-au atins



obiectivele stabilite pentru aplicații cu caracter limitat, bine definite, cum ar fi în domeniul sistemelor bazate pe cunoștințe și al sistemelor cu limbaje naturale restrînse la o tehnologie specifică. S-au constituit peste 50 de sisteme bazate pe cunoștințe, dintre care cele mai importante se prezintă în tabelul 1.2.

TABELUL 1.2. **Sisteme bazate pe cunoștințe**

| Numele | Funcția sistemului | Firma producătoare |
|---|---|---|
| 1 | 2 | 3 |
| ACE | Mentenabilitatea cablurilor telefonice | AT & T Bell Labs. |
| ADA-Tutor | Învățarea programării Ada | Computer Thought |
| Airplan | Decolarea și aterizarea avioanelor | U. S. Navy |
| Dendral | Analiza în spectrometria de masă | Universitatea Stanford |
| Dip Meter Advisor | Analiza zăcămintelor de petrol | Schlumberger Ltd. |
| Drilling Advisor | Diagnosticarea forării petrolului | Teknowledge |
| Expert-Ease | Luarea deciziilor — calculatoare IBM-PC | Export Software International Ltd. |
| Explorer | Interfața utilizatorului la programe geologice complexe | Cognitive Systems |
| Induce/Plant | Diagnosticarea bolilor la soia | Universitatea din Illinois |
| IPP | Înțelegerea știrilor | Universitatea Yale |
| CATS-1 | Depanarea locomotivelor | General Electric |
| Macsyma | Rezolvarea formulelor matematice | MIT |
| Molgen | Experimentări în biologia moleculară | Universitatea Stanford |
| Mycin | Diagnosticarea bolilor sanguine | Universitatea Stanford |
| Oncocin | Tratamentul bolii Hodgkins | Universitatea Stanford |
| Prospector | Exploatarea zăcămintelor de minereuri | SRI International |
| Puff | Diagnosticarea maladiilor pulmonare | Universitatea Stanford |
| SMP | Efectuarea operațiilor matematice simbolice | Inference Corp. |

| 1 | 2 | 3 |
|---|---|---|
| X-CON | Stabilirea configurației calculatoarelor pentru fabricație | Universitatea Carnegie Mellon |
| X-SEL | Stabilirea configurației calculatoarelor pentru comercianți | DEC |

Sistemele care folosesc limbaje naturale asigură interfața între o persoană și un program complex de calculator prin utilizarea limbii vorbite. Au fost dezvoltate cîteva programe care utilizează limbaje naturale pentru asigurarea unei interfețe utilizator — baze de date. De fapt, prima aplicație practică de inteligență artificială a fost o asemenea interfață cu sisteme de managament al bazelor de date, organizate pe calculatoare IBM, cunoscută sub numele *Intellect.*

Cercetările în domeniul evaluării asistate de calculator a limbajelor naturale au o istorie proprie care începe în anii '50, odată cu inițierea primelor programe de traducere automată. Primele încercări se bazau pe găsirea sinonimelor și expresiilor idiomatice fără a se ține seama de context, ceea ce a condus la rezultate slabe.

Una dintre cele mai vechi tehnici de inteligență artificială pentru crearea mașinii inteligente este cunoscută sub numele de *examinarea spațiului stărilor ;* ea se bazează pe trecerea în revistă a tuturor stărilor posibile în care se poate afla un sistem și pe descrierea fazelor intermediare de trecere de la faza inițială la cea finală. Această aproximație a fost folosită în situații bine definite, cum ar fi jocul de șah. Claude Shannon, întemeietorul teoriei informației, a calculat în anul 1950 că un program de șah are $10^{120}$ mutări pentru o partidă, dacă se iau în considerare toate posibilitățile, ceea ce ar însemna un timp de joc practic infinit. Concluziile desprinse au condus la diversificarea metodelor folosite și la dezvoltarea rapidă a sistemelor de limbaje naturale (tabelul 1.3.).

Ca și în celelalte discipline științifice, inteligența artificială include cîteva arii distincte de cercetare, fiecare cu interese, tehnici și terminologii specifice ; astfel, pot



TABELUL 1.3. **Sisteme de limbaje naturale**

| Numele | Funcția sistemului | Firma producătoare |
|---|---|---|
| Alps | Traducere interactivă | Alps Inc. |
| Ask | Sistem de management al bazei de date cu limbaj natural | Caltech |
| Epistle | Prelucrarea textelor în limbaj natural | IBM |
| Intellect | Interfață cu sisteme de management al bazei de date cu limbaj natural | AI Corp. |
| Logos | Traducerea automată a limbajelor | Logos Computer System Inc. |
| Natural Link | Sistem de management al bazei de date cu limbaj natural | Texas Instruments |
| Pearl | Interfață în limbaj natural pentru baze de cunoștințe | Cognitive Systems Inc. |
| Savvy | Interfață în limbaj natural pentru calculatoare personale | Excalibur Technologies Corp. |
| Straight Talk | Interfață în limbaj natural pentru sisteme de prelucrare a cuvintelor | Dictaphone Corp. |
| Team | Mecanism de acces cu limbaj natural la sisteme de management al bazei de date | SRI International |
| Themis | Sistem de management al bazei de date cu limbaj natural pentru VAX-Oracle | Frey Associates |
| Weidner System | Traducerea semiautomată | Weidner Communication Corp. |
| Ham-Ans | Interfață de uz general la sisteme de management al bazei de date | Universitatea Hamburg |
| Supernatural | Interfață cu limbaj natural pentru prelucrarea informațiilor | Microdata |

fi identificate următoarele domenii : înțelegerea vorbirii, sisteme vizuale, mijloace și limbaje de programare, rezolvarea problemelor, programarea automată etc. În cele ce urmează se prezintă unele dintre aceste arii de interes, cu precizarea că nu este posibilă separarea lor, majoritatea cercetătorilor fiind implicați practic în toate aspectele privind inteligența artificială.

Elaborarea modalităților de soluționare a problemelor este legată de succesele obținute de programele destinate

simulării unor jocuri, cum ar fi șahul. Acestea folosesc tehnici de căutare sau reducere, constînd în prefigurarea unor mutări viitoare în joc sau, respectiv, divizarea în subprobleme. Pînă în prezent nu s-au găsit încă soluții în ceea ce privește capacitatea omului de a analiza o configurație în ansamblu și, de asemenea, posibilitatea conceptualizării, aspect care în domeniul inteligenței artificiale a fost denumit *alegerea reprezentării problemei;* totodată modul de a gîndi asupra rezolvării implică pentru programe un spațiu în care se caută soluția.

Deducțiile logice au jucat de la început un rol important deoarece programele trebuie să realizeze legături prin utilizarea unei baze de date, conținînd pentru fiecare aspect analizat structuri discrete de date, în mod asemănător cu reprezentarea prin formule discrete — modalitate folosită în logica matematică. Aria gîndirii logice este una dintre cele mai investigate din inteligența artificială.

În prezent se caută soluții pentru concentrarea asupra unor fapte relevante dintr-o bază mare de date păstrarea elementelor legate de anumite aserțiuni, precum și completarea acestora la primirea de informații noi.

Înțelegerea vorbirii reprezintă o latură importantă a inteligenței artificiale, fiind realizate programe pentru traduceri sau acumularea de cunoștințe din texte și constituirea unei baze de date interne. Principalele aspecte privind înțelegerea vorbirii se referă la reprezentarea importanței cunoștințelor cu sens comun și a rolului presupunerilor legate de subiectul în discuție pentru interpretarea propozițiilor. În momentul de față există o varietate destul de mare de programe cu posibilități de înțelegere a vorbirii, dar limitate în general la anumite domenii.

O arie deosebită de cercetări în domeniul inteligenței artificiale o constituie programarea automată pornind de la descrierea scopului, algoritmului etc.; o varietate importantă o reprezintă programele care se autoinstruiesc, modificîndu-și comportarea prin schimbarea codurilor proprii. Realizările legate de autoinstruire nu sînt semnificative.



Sistemele-expert sau ingineria cunoașterii au evoluat pînă la nivelul aplicațiilor utile, situîndu-se din acest punct de vedere printre cele mai avansate tehnici ale inteligenței artificiale. Utilizatorul lucrează în modul interactiv, explicînd problema, executînd testele ce i se sugerează și punînd întrebări asupra soluțiilor propuse. Sistemele-expert se înscriu în spațiul dintre experții umani, care interacționează cu sistemele în modul achiziționarea de cunoștințe, și utilizatori, care interacționează cu sistemele în modul consultație. În prezent se caută noi posibilități pentru ca sistemele-expert să-și explice modul de gîndire.

În domeniul inteligenței artificiale cercetarea este orientată și spre elaborarea de programe care comandă manipularea dispozitivelor roboților, de la mișcarea optimă a brațelor pînă la metodele de planificare a secvențelor de acțiuni pentru atingerea scopurilor impuse. Prelucrarea informației vizuale reprezintă în ciuda unor dificultăți o arie deosebit de investigată de specialiștii în domeniul inteligenței artificiale. Pînă în prezent au fost elaborate programe care permit recunoașterea obiectivelor și umbrelor în scenele vizuale și chiar sesizarea diferențelor între imagini, cum ar fi, de exemplu, fotografiile aeriene.

Limbajele de reprezentare a cunoștințelor — de fapt limbaje pe calculator pentru codificarea cunoștințelor și a metodelor de gîndire ca structuri de date și proceduri— au fost dezvoltate în scopul materializării ideilor referitoare la realizarea unor programe deductive.

Evoluții semnificative vor avea loc în domeniul sistemelor hardware destinate activităților de inteligență artificială, accentul punîndu-se pe echipamente care să lucreze în paralel.

O mașină de inferențe în paralel va atinge 100 milioane ÷ 1 miliard de inferențe logice pe secundă (LIPS — *L*ogical *I*nferences *P*er *S*econd). Pe un calculator actual o inferență are între 100 și 1 000 pași; rezultă că 1 LIPS este echivalent cu 100 ÷ 1 000 instrucțiuni/s. O mașină bazată pe cunoștințe este un sistem de calcul care manipulează cantități mari de cunoștințe — date și proceduri — și organizează, controlează, reface și actuali-

zează automat conținutul său. Capacitatea sa va fi de 10 miliarde — 100 miliarde octeți ; aceasta va include funcții de inferențe ca și funcții și relații algebrice.

Cele două mașini — de inferențe și, respectiv, bazate pe cunoștințe — vor fi unite pentru a se realiza un calculator complet din generația a cincea ; în combinație, acestea vor fi capabile să caute date relevante și să găsească, în cîteva secunde, soluții la probleme complexe. Mașina bazată pe cunoștințe va fi capabilă să funcționeze ca o mașină convențională de bază de date.

Indiferent de forma și funcția lor, toate calculatoarele din generația a cincea vor avea în esență aceeași structură — o mașină de inferențe în paralel cu software-ul de management bazat pe cunoștințe.

Aspectele menționate implică, pentru sfîrșitul deceniului actual și începutul deceniului următor, o imagine complet nouă asupra tehnicii de calcul.

La începutul anilor '90 vor deveni posibile aplicații inedite, presupunînd că aproape fiecare dintre noi am putea fi utilizatorul calculatoarelor electronice din cea de-a cincea generație.

Automatizările industriale vor beneficia într-o măsură foarte mare de noua generație de calculatoare. În prezent sînt în funcțiune roboți, într-o gamă impresionantă, iar tehnicile de proiectare asistată de calculator (CAD — *C*omputer *A*ided *D*esign) și tehnicile de fabricație asistată de calculator (CAM — *C*omputer *A*ided *M*anufacturing) au fost introduse pe scară largă de către firme din cele mai diverse domenii. Prin integrarea roboților și a sistemelor CAD și CAM s-a trecut la automatizarea completă a întreprinderilor industriale.

Calculatoarele din generația a cincea vor avea efecte atît asupra liniilor cu capacitate mare de producție, cît și asupra celor de serie mică sau unicate prin utilizarea roboților dotați cu sisteme vizuale computerizate, de recunoaștere și sinteză a vorbirii, cu limbaje de programare de nivel înalt etc. De asemenea se vor crea condiții pentru înlocuirea muncii omului în medii toxice sau cu un grad înalt de periculozitate, ca de exemplu în tehnica reactoarelor nucleare.



În domeniile atît de diverse ale științei și tehnologiei cercetătorii, proiectanții și inginerii vor folosi calculatoarele din noua generație pentru memorarea, căutarea și redobîndirea informațiilor, de la documentații științifice păstrate în baze de date de tipul bibliotecilor, pînă la date experimentale obținute în timp real de la instrumente științifice conectate la calculatoare. Procesoare paralele pe scară largă vor fi utilizate pentru analiza datelor colectate de rețele vaste de calculatoare de capacitate mică. Pe baza analizei pe scară largă a activității seismice înregistrate în timp real de la stații îndepărtate ar putea fi posibilă prevederea cutremurelor de pămînt. Ca aplicații ale calculatoarelor electronice din generația a cincea mai semnalăm sistemele-expert pentru diagnosticarea defectelor, aplicarea expertizelor științifice, interfețele inteligente om-mașină pentru simularea proceselor fizice cu calculatoare obișnuite, instrumentele analitice îmbunătățite, cum ar fi programele de algebră simbolică, bazele integrate de date științifice și tehnologice, sistemele-expert pentru analize geologice și prospecțiuni de minereuri — echipamente care au început să fie folosite în diferite cercetări științifice. Într-adevăr, zăcămintele pot fi localizate și analizate de către geologi cu programe de prelucrare simbolică. În mod similar a crescut eficiența simulării prin metode numerice complexe a structurilor petrolifere.

Proiectarea semiautomată și dezvoltarea programelor pentru calculatoare electronice (hardware și software), sistemele-expert pentru verificarea și întreținerea programelor, arhitecturile specializate pentru implementarea microinstrucțiunilor (*firmware)* și pentru diagnoza defectelor hardware la calculatoare cu limbaje de programare de nivel foarte înalt, procesoarele paralele pentru analiza și proiectarea circuitelor VLSI (*V*ery *L*arge *S*cale *I*ntegration), stațiile de lucru de înaltă performanță pentru dezvoltarea software, limbajele paralele de programare și sistemele de operare concură la reducerea costurilor de producere a software-ului.

Programarea automată, completă, de uz general va necesita mult mai multe eforturi din punctul de vedere al cercetării fundamentale. Investigațiile în domeniul

limbajelor de programare destinate prelucrării datelor — VAL, FGL și PROLOG — ca și în domeniul elaborării echipamentelor destinate transmiterii și prelucrării datelor ar putea face posibilă programarea calculatoarelor paralele cu un număr mare de procesoare paralele fără a fi necesară intervenția programatorului. În prezent există diferite instrumente deosebit de utile în programare — verificatoare, editoare și dispozitive de întreținere a fișierelor. Unele proiecte încearcă să combine modalitățile de procesare în paralel pe scară largă pentru analiza circuitelor cu sisteme CAD și sisteme-expert pentru proiectarea VLSI, în principal la conexiunile și organizarea *(layout*-ul) circuitelor. Simularea și proiectarea asistată de calculator vor deveni aplicații de bază ale noii generații de calculatoare.

În medicină se apreciază că sistemele-expert vor continua să cîștige teren, unele dintre acestea fiind utilizate cu succes în diagnosticarea diferitelor maladii. Multe din problemele de medicină implică analize de diagnosticare clasice pe care sistemele-expert le rezolvă în bune condiții. În plus trebuie să avem în atenție faptul că în medicină, ca de altfel și în alte domenii de activitate, există numeroase arii de investigare, unde accesul experților umani este dificil și costisitor. Totodată costul ridicat al metodelor clasice folosite în medicină justifică eforturile financiare privind constituirea bazelor de cunoștințe pentru sistemele-expert, care pot determina reducerea cheltuielilor cerute de realizarea actualei aparaturi medicale destinate diagnosticării. Tehnologia folosită pentru realizarea calculatoarelor din generația a cincea poate sta la baza construirii protezelor pentru handicapați și invalizi. În prezent există deja mașini de citit pentru orbi, iar cercetările asupra senzorilor și cărucioarelor controlate cu calculatorul au loc în mai multe laboratoare din diferite țări. Menționăm și faptul că printre aplicațiile noii generații de calculatoare se află brațele robotizate utilizate pentru ajutorarea invalizilor.

Instruirea asistată de calculator (CAI — *C*omputer *A*ided *I*nstruction) se folosește în mod curent în învățămîntul de toate treptele. Datorită puterii sporite de calcul,



realizării unor interfețe tot mai perfecționate și a calculatoarelor personale „multimedia", generația nouă de calculatoare va face CAI disponibilă pentru grupuri din ce în ce mai largi de utilizatori. Rețele de calculatoare vor asigura accesul la o gamă largă de cursuri și vor furniza beneficiarilor multiple servicii, dintre care menționăm teleconferințele. Sistemele mai evoluate vor permite studenților să folosească limbajul natural și comenzile prin voce pentru a controla fluxul informațiilor atît ca secvență, cît și ca grad de detaliere. Cercetări în domeniul psihologiei, orientate spre modelarea gîndirii umane și a comportamentului în rezolvarea problemelor, ar putea contribui la îmbunătățirea sistemelor CAI prin adaptarea acestora la diferite stiluri de a învăța și la cunoștințele inițiale ale studenților. În prezent există calculatoare-asistent rudimentare care aplică modul de lucru constînd în utilizarea și diagnosticarea problemelor în interdependență cu alte sisteme— un program de aplicație pentru calculator, un automobil, un dispozitiv electronic sau un manual de proceduri de operare etc.

În comerț și servicii, consumatorii vor putea face cumpărături utilizînd cataloage implementate pe calculator care vor asigura atît descrierea textuală, sonoră, grafică și imaginea produselor solicitate, cît și calculul, plata și livrarea mărfurilor. O „carte" similară pe calculator, susținută de un sistem-expert și o interfață pentru limbaj natural, va permite localizarea imediată a magazinelor, iar îmbunătățirea sistemelor de recunoaștere individuală a vocii vor asigura efectuarea cumpărăturilor prin telefon la comandă verbală. Aplicații multiple vor apare în sfera serviciilor — planificarea deplasărilor și excursiilor, analize financiare, informații asupra pieței, accesul la știri și rapoarte meteorologice la cerere, precum și diagnosticarea și depistarea defectelor posibile ale unor bunuri de uz gospodăresc.

Piloții, controlorii traficului aerian, specialiștii care se ocupă de studiul și exploatarea spațiului cosmic precum și cu detectarea de la distanță a resurselor Terrei vor utiliza calculatoarele din noua generație. În viitor tehnicile folosite de noua generație de calculatoare — re-

cunoașterea vocii, sinteza vorbirii și sistemele-expert — vor ajuta personalul menționat să selecteze și să interpreteze cele mai importante informații. Sistemele de vizualizare conectate la un calculator vor permite ghidarea zborurilor prin compararea continuă a hărților digitalizate cu imaginile terenului survolat. Programe îndrăznețe au fost concepute pentru depistarea din sapțiu a resurselor terestre, precum și pentru descoperirea și exploatarea bogățiilor de pe alte planete. Astfel de aplicații la care se adaugă monitorizarea fenomenelor meteorologice la scară globală și a resurselor Pămîntului, vor necesita roboți de complexitate mărită, înzestrați cu multiple posibilități.

În domeniul militar calculatoarele din noua generație vor putea fi folosite atît ca sisteme pentru managementul informațiilor, cît și ca sisteme-expert pentru asistarea sistemelor militare în ceea ce privește planificarea, luarea deciziilor și diagnosticarea defectelor. Calculul simbolic și hardware-ul din noua generație vor deține un rol important în comunicații, comandă și control. Arhitecturile paralele pentru analiza imaginilor de radiolocație și sonar, precum și pentru planificare și desprinderea în mod automat a concluziilor vor fi încorporate în sistemele cu destinație militară, element ce va implica inclusiv reproiectarea și dotarea armamentului cu interfețe specifice.

Alte aplicații ale calculatoarelor din generația a cincea vizează automatizarea sistemelor informaționale din birouri (acestea vor fi dotate cu calculatoare personale, de mare putere și cu rețele largi conectate la sisteme de calcul centrale), înregistrarea digitală și memorarea lucrărilor din domeniile muzicii, picturii și artei grafice, realizarea unor roboți casnici inteligenți etc.

## 1.3. TEHNICI PENTRU REZOLVAREA PROBLEMELOR

În domeniul inteligenței artificiale noțiunile de *rezolvarea problemelor* și *examinare* se referă la o serie de principii care privesc deducțiile, inferențele, raționamentele comune, verificarea teoremelor și procesele asociate. Aplicații ale acestor principii se găsesc în programele pentru înțelegerea limbajelor naturale, restabilirea informației, programarea automată, analiza imaginilor, jocuri, sistemele-expert și verificarea teoremelor matematice. Modalitatea principală folosită pentru rezolvarea problemelor o constituie tehnicile de examinare ; elementele de bază ale acestora sînt date de modul de reprezentare a problemelor. Astfel, sînt cunoscute reprezentări în spațiul stărilor, prin reducerea problemelor, precum și sub forma unor arbori, modalitate care a fost utilizată încă de la început în teoria jocurilor pe calculatorul electronic.



În sistemele de rezolvare a problemelor componentele principale sînt baza de date, operatorii și strategia de control.

*Baza de date* este alcătuită din liste, expresii, structuri și rețele care descriu starea fiecărei etape parcurse. Pentru exemplificare, în verificarea teoremelor baza de date este constituită din axiome, leme și teoreme cunoscute utilizate pentru atingerea scopului propus, iar în aplicațiile roboților, din modelul operațiilor pe care aceste dispozitive trebuie să le realizeze, scopul fiind descrierea secvenței de urmat în mișcare.

*Setul de operatori*, ca de exemplu regulile pentru inferențe în verificarea teoremelor sau a regulilor jocului în șah, contribuie la generarea stărilor următoare în procesul de căutare a soluției. În cazul inteligenței artificiale, de cele mai multe ori se utilizează un set redus de operatori pentru generarea celorlalți prin inferențe.

*Strategia de control* implică luarea unei decizii privind pasul următor starea asupra căreia se va acționa, precum și operatorul cel mai indicat. Conținutul și structura bazei de date sînt în general dependente de strategia de control aleasă. Reprezentările posibile conduc la strategii de control, paralele sau alternative, în funcție de modul de abordare a secvenței operatorilor.

Modul de aplicare a operatorilor asupra bazei de date este diferit, fiind determinat de direcția de acțiune —

înainte sau înapoi. Atingerea scopului final pornind de la starea inițială se face prin raționament înainte, direct sau jos-sus și se bazează pe un anumit tip de operatori (fig. 1.1). Aplicarea operatorilor pentru extragerea din

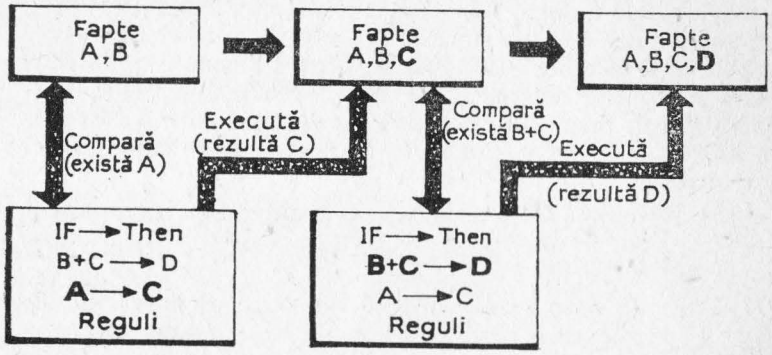



*Fig. 1.1.* Raționamentul direct.

starea finală a unor probleme intermediare, mai ușor de rezolvat, determină un raționament înapoi, invers sau sus-jos (fig. 1.2).

Cele mai multe tehnici de programare folosite în domeniul inteligenței artificiale se bazează pe raționamentul invers, care este mult mai apropiat de modul de ju-

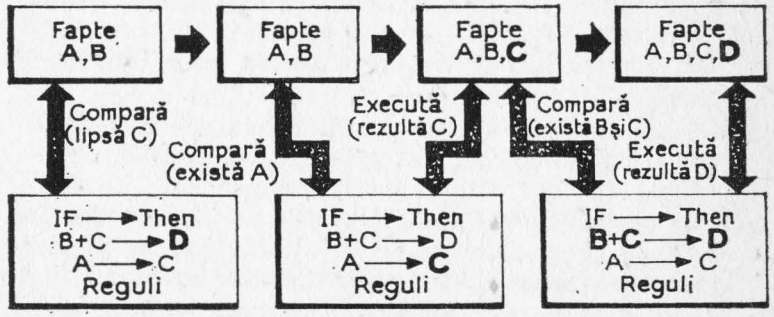



*Fig. 1.2.* Raționamentul invers.



decată uman. Există și posibilitatea folosirii combinate a raționamentului direct și invers, ceea ce implică mai întîi sesizarea deosebirilor între două stări, iar apoi aplicarea unui operator de tipul sus-jos ; în cazul apariției unor dificultăți se acționează pentru găsirea unor stări intermediare, după care se utilizează operatorul potrivit.

Reprezentarea în spațiul stărilor face apel la unități distincte, cum ar fi starea — definită ca un element de structură arătînd situația atinsă în fiecare etapă pe calea urmată pentru găsirea soluției — și operatorii — funcții ce reflectă transformările de la o stare la alta. Soluția unei probleme este căutată prin aplicarea setului specific de operatori stării inițiale pentru a se atinge scopul propus. Spațiul stărilor poate fi înfățișat sub forma unui graf, ca în fig. 1.3., în care nodurile sînt stările, iar arcele operatori. În cazurile în care problema are mai multe stări finale, de cele mai multe ori se cere aflarea soluției prin impunerea condiționării sub forma unui minim (lungime, cost etc.).

Sistemele de rezolvare a problemelor prin reprezentare în spațiul stărilor utilizează raționamentul direct pentru a produce o nouă stare în baza de date. În spațiul stărilor poate fi folosit totuși și raționamentul invers, în cazul în care fiecare aplicare a unui operator conduce la o singură problemă nouă.

Tehnica reprezentării prin reducerea problemei implică descrierea inițială a stării finale și transformarea acesteia într-un set de subprobleme, denumite și probleme primitive care au soluții imediate. În acest caz

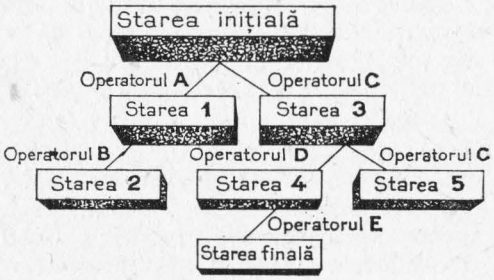



*Fig. 1.3.* Graf pentru reprezentarea în spațiul stărilor.

se utilizează raționamentul invers, aplicarea unui operator conducînd la divizarea problemei într-un set care poate fi rezolvat mai ușor. Pentru exemplificare, în fig. 1.4 se prezintă un graf ȘI/SAU specific acestui tip de problemă.

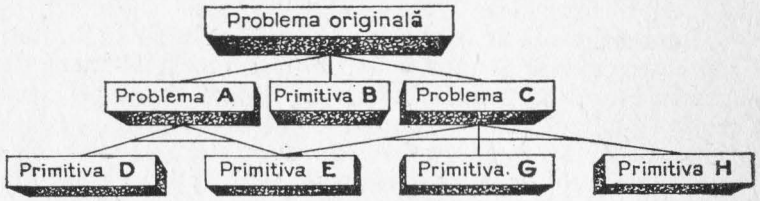



*Fig. 1.4.* Graf ȘI/SAU pentru reprezentarea prin reducerea problemei.

Regulile folosite pentru construcția grafului ȘI/SAU sînt următoarele :

— primul nod corespunde problemei originale ;

— nodurile intermediare reprezintă fie o singură problemă, fie un set de probleme ;

— nodurile terminale, fără descendenți, reprezintă primitive ;

— problema poate fi rezolvată dacă cel puțin unul din nodurile SAU (în cazul nostru *A, B* sau *C)* are soluție ;

— problema poate fi rezolvată dacă toate nodurile ȘI (în cazul ilustrat în fig. 1.4 fie *D* și *E,* fie *E, G,* și *H)* au soluție ;

Uneori se utilizează ambele tehnici de reprezentare descrise mai sus, cu precizarea că trecerea de la reducerea problemelor la spațiul stărilor este ceva mai complexă în principal atunci cînd există noduri ȘI.

Pentru realizarea de programe pe calculatorul electronic destinate simulării jocurilor — de dame, șah, Go etc., în care de obicei sînt doi parteneri — este necesară reprezentarea tuturor posibilităților de mutare, sub forma unui graf de tip arbore. Diferența principală dintre arborele unui joc și reprezentarea în spațiul stărilor



este dată de numărul de participanți — doi și, respectiv, unul.

Dificultatea esențială în cazul arborilor este numărul imens de posibilități. Astfel, A. L. Samuel a demonstrat că la jocul de dame pot exista pînă la $10^{40}$ moduri intermediare, ceea ce ar însemna, la o viteză de generare de trei miliarde noduri pe secundă, un timp de $10^{21}$ secole ; pentru șah C. E. Shannon a găsit un număr de $10^{120}$ posibilități, în cazul unor jocuri de durată medie.

În afară de cazurile descrise mai sus, în care fiecare operator pleacă de la o singură stare, există și baze de date pentru care operatorii au mai multe intrări și o singură ieșire, cum ar fi, de exemplu, reprezentarea teoremelor.

Probleme tipice, ușor de reprezentat în tehnicile descrise, sînt verificarea teoremelor în logică și matematică, găsirea drumului minim, alegerea mutărilor bune pentru cîștigarea unui joc, integrarea simbolică prin transformări succesive etc.

Pentru reprezentarea problemelor se folosește o serie de algoritmi de examinare a spațiului în care se poate găsi soluția. Căutarea se face prin investigarea unei părți a grafului, suficient de mare pentru a conține un traseu care reprezintă o soluție. Dacă ordinea investigării este arbitrară, adică nu se analizează informațiile disponibile pentru orientarea cercetării, metoda este denumită examinare oarbă. Procedura de extindere a grafului de la un anumit nod la nodul următor, aplicînd un singur operator, diferă de la un caz la altul ; în acest fel rezultă mai mulți algoritmi de examinare oarbă : la prima vedere, a costurilor uniforme, în adîncime și bidirecțională.

Primele trei metode folosesc raționamentul direct și se bazează pe următoarele ipoteze :

— graful spațiului stărilor este un arbore, ceea ce implică existența unui singur nod inițial și a unor trasee unice între noduri ;

— la fiecare extindere spre nodurile următoare se înregistrează locul de plecare pentru a se putea recunoaște imediat traseul urmat în cazul găsirii soluției.

Algoritmul de examinare *la prima vedere* constă în determinarea distanței dintre un nod oarecare și nodul inițial, măsurată prin numărul arcelor existente între nodurile respective.

De obicei se face o diferențiere între graful complet al problemei — denumit *spațiul de examinare* — și graful sau arborele care se construiește pe măsura avansării spre nodul final, cunoscut sub numele de *graful* sau *arborele de examinare*. Pentru multe probleme spațiul de examinare este infinit sau foarte mare, în timp ce pentru găsirea unei soluții de cele mai multe ori se explicitează numai o parte limitată a grafului.

Prin generalizarea algoritmului examinării *la prima vedere* se ajunge la metoda *costurilor uniforme*, caz în care fiecărui arc îi corespunde un anumit cost. Problema principală constă în găsirea unui traseu al costurilor minime.

Examinarea *în adîncime* se referă la extinderea în continuare a nodului succesor, generîndu-se în acest fel un singur traseu în spațiul stărilor. În multe probleme se fixează adîncimea examinării pentru a nu se utiliza un drum de lungime prea mare. Pentru exemplificare, se poate considera cazul analizei unei mutări posibile în jocul de șah, pentru care *adîncimea* se fixează sub forma unui număr limitat de răspunsuri „văzute" de parteneri.

Uneori se utilizează raționamentul invers — de la nodul final spre nodul inițial. Prin combinarea raționamentului direct cu cel invers se obține tehnica examinării *bidirecționale*. Metoda descrisă constă în generarea a două grafuri de examinare : unul care pleacă de la nodul inițial, iar cel de-al doilea de la nodul final. Generarea are loc pînă la intersectarea celor două grafuri.

Algoritmii construiți pe baza reprezentării prin reducerea problemei conduc la generarea unor grafuri ȘI/SAU definite printr-un nod inițial sau descrierea problemei, un set de noduri terminale sau descrierea primitivelor și un set de operatori pentru reducerea problemei la subprobleme. Spre deosebire de majoritatea algoritmilor de examinare în spațiul stărilor, metodele bazate pe grafuri ȘI/SAU utilizează raționamentul invers și se aplică ținînd seama de următoarele ipoteze :



— spațiul de examinare este un arbore ȘI/SAU și nu un graf general ;

— subproblemele pot fi rezolvate în ordine arbitrară.

Categoriile în care se împart metodele de mai sus depind de arborii ȘI/SAU realizați prin examinarea la prima vedere sau în adîncime.

Toți algoritmii descriși mai sus generează un număr foarte mare de noduri într-o ordine arbitrară și nu țin seama de informațiile sau proprietățile specifice problemei de rezolvat, rezultînd practic o explozie de combinații în spațiu și timp pînă la găsirea soluției. Reducerea numărului de noduri se poate face folosind mai multe modalități :

— transpunerea problemei pentru micșorarea mărimii spațiului de examinat ;

— utilizarea înțelegerii euristice pentru concentrarea investigării și eliminarea multitudinii de combinații ; aceasta reprezintă una din cele mai importante contribuții ale inteligenței artificiale la rezolvarea eficientă a problemelor ;

— programarea pentru limitarea examinării ; în acest sens s-au introdus macrooperatori și conceptul de planificare pentru renunțarea la detaliile nesemnificative.

Diferența dintre examinarea oarbă și cea euristică constă în faptul că, în primul caz elementele spațiului stărilor sînt generate și verificate sistematic, lăsînd loc însă și unor informații suplimentare, care nu sînt excluse prin definiție. Informațiile introduse pot dovedi limitarea examinării și, în funcție de aceasta, metoda este fie euristică, fie oarbă. Cu alte cuvinte, informațiile euristice sînt cele suplimentare în raport cu necesarul minim pentru definirea stărilor și operatorilor.

Inițial, majoritatea programelor pentru rezolvarea problemelor utilizau metodele euristice de examinare ; aceste programe erau însă strîns legate de tehnicile de lucru specifice domeniului analizat, ceea ce a împiedicat dezvoltarea și adaptarea lor la probleme noi. Ca urmare s-au efectuat cercetări intense pentru definirea unor algoritmi euristici generali care să poată fi studiați în mod independent.

De cele mai multe ori pentru caracterizarea tehnicilor euristice se pleacă de la metodele de examinare oarbă. Principalele elemente care folosesc informațiile euristice sînt următoarele : luarea deciziei privind direcția de extindere a nodurilor — operație care nu se mai face în mod aleator —, generarea nodului descendent în funcție de decizia luată, precum și eliminarea unor anumite noduri, care nu vor mai fi incluse în arborele examinării.

Decizia asupra direcției de extindere utilizează un algoritm de examinare a spațiului stărilor în care se presupune, că nodurile se generează complet sau deloc, urmîndu-se calea cea mai promițătoare ; aceasta este așanumita examinare comandată, ordonată sau a celui mai bun descendent.

Selectarea operatorului ce trebuie aplicat determină nodul descendent posibil de generat. Nodurile cărora le sînt aplicați numai unii operatori, urmînd ca examinarea completă să se facă ulterior, au fost denumite *noduri dezvoltate parțial* sau *extinse parțial*. Uneori limitarea descendenților se face prin alegerea operatorului aplicabil imediat pentru avansarea examinării. Eliminarea nodurilor nepromițătoare are loc atît pentru economisirea spațiului, care altfel ar fi destinat inutil cercetării, cît și pentru restrîngerea examinării.

Definirea unui nod promițător se face fie prin calculul distanței față de un nod final, mai ales pentru problemele în spațiul stărilor, fie prin analiza lungimii sau dificultății drumului care cuprinde acel nod. Nodurile sînt estimate cu ajutorul unei funcții de evaluare, $f^x$ care are o valoare cu atît mai mică cu cît nodul este mai promițător. Pentru exemplificare, în cazul examinării la prima vedere se poate alege ca funcție de evaluare adîncimea nodului ; dacă pentru examinare se folosește metoda costurilor uniforme, atunci funcția de evaluare este costul pînă la nodul dat, iar în situația examinării în adîncime, $f^x$ este valoarea negativă a adîncimii nodului.

Aplicabilitatea funcției de evaluare depinde de tipul soluțiilor cerute de problema studiată. De cele mai multe ori se caută o soluție optimă folosind așa-numitul algoritm $A^x$ ; în cazul problemelor complexe se examinează



soluțiile pentru care raportul dintre efortul cercetării și gradul de precizie se apropie de un minim. Atunci cînd nu se pune problema unei soluții optime — deoarece, de cele mai multe ori, soluția este singulară — se caută ca efortul de examinare să fie minim.

Dacă pentru analiza grafurilor ȘI/SAU se folosește metoda examinării euristice, atunci nu mai există o corespondență biunivocă între posibilitățile de extindere ale unui nod și cele ale unei soluții potențiale.

Pentru examinarea arborilor jocurilor se utilizează tehnici de tipul minimax, alfa-beta sau euristice.

Pentru aplicarea metodei minimax la jocurile cu doi parteneri, *1* și *2*, pentru care pozițiilor terminale li se alocă anumite valori, calculul unei faze intermediare se face după cum urmează :

— valoarea unui nod cu ascendent SAU, din care partenerul *1* alege următoarea mutare, este pentru *1* maximul oricăruia dintre descendenți ;

— valoarea unui nod cu descendenți ȘI, din care partenerul *2* alege următoarea mutare, este pentru *1* minimul oricăruia dintre descendenți.

Un algoritm similar este determinat de formalismul negmax, caz în care valoarea alocată fiecărui nod este cea dată de jucătorul aflat la mutare, cu precizarea că valorile corespunzătoare celor doi parteneri diferă prin semn. Mutarea cea mai bună pentru oricare din parteneri corespunde nodului cu valoare maximă.

Pentru elaborarea variantelor descrise mai sus s-a presupus că arborele jocului a fost generat complet. Pentru simplificare se poate genera numai o anumită porțiune a arborelui plecînd de la poziția curentă, ceea ce înseamnă de fapt toate mișcările cu o limită fixată a adîncimii, spre exemplu. Procedura minimax aplicată unui arbore generat parțial face apel la o funcție de evaluare, avînd un scop similar cu cel al funcției euristice utilizate la examinarea comandată sau a arborilor ȘI/SAU. Dacă arborele parțial al jocului conține noduri terminale, funcția statică de evaluare are valoarea plus infinit la cîștig, minus infinit la pierdere și zero în caz de egalitate.

Tehnica alfa-beta utilizează doi parametri, alfa și beta, reprezentînd valoarea maximă a numărului conexiunilor inferioare, respectiv, superioare. Procedura garantează că nodul inițial al arborelui va avea aceeași valoare finală, ca și în cazul folosirii formalismului minimax complet. Gradul în care procedura alfa-beta reprezintă o îmbunătățire față de metoda minimax depinde de ordinea în care sînt evaluate nodurile. În general, este de dorit ca prima dată să fie evaluat cel mai bun descendent al fiecărui vîrf ascendent. În acest sens o posibilitate constă în aplicarea funcției de evaluare statică la descendenți și în considerarea rezultatelor acestei evaluări preliminare ca o aproximare a valorilor așteptate.

Deoarece procedura alfa-beta este mai complicată decît cea de tip minimax, rezultatul fiind același, s-a ajuns la arbori în care nodurile extreme au aceeași adîncime, iar celelalte generează un număr egal de descendenți.

Examinarea arborilor jocurilor se poate face utilizînd, în anumite puncte, informații euristice. Acest lucru este valabil mai ales atunci cînd numărul variantelor posibile este mare. O asemenea situație o întîlnim la jocul de șah, unde nu se poate aștepta pînă la găsirea mutării care duce forțat la cîștig — dacă o astfel de mutare există (!), — ci se preferă o mutare care poate fi calificată drept „bună".

Majoritatea programelor de șah pentru calculatoare electronice folosesc metoda de examinare propusă de Shannon încă din anul 1950. Denumită de tip A, această metodă se caracterizează prin aceea că arborele jocului este generat numai pînă la o anumită adîncime. Întrucît nodurile respective nu sînt terminale se definește o funcție pentru evaluarea lor. În varianta de tip B, dezvoltată ulterior, examinarea se extinde la pozițiile de joc cu stabilitate mică, pentru care s-au luat în considerare toate posibilitățile.

Cercetări mai noi au condus la elaborarea unor programe bazate pe cunoștințe organizate într-o manieră apropiată de gîndirea umană, care urmăresc generarea selectivă a mutărilor, în funcție de situația de pe tabla de joc.



Dintre cele mai cunoscute programe de examinare menționăm : teoreticianul logic *(Logic Theorist)*, rezolvitorul de probleme generale *(General Problem Solver)*, mașina de verificare a teoremelor din geometrie *(Geometry Theorem-proving Machine)*, integratorul simbolic automat — SAINT (*S*ymbolic *A*utomatic *INT*egrator) —, STRIPS și ABSTRIPS.

Teoreticianul logic a fost scris de A. Newell, J. C. Shaw și H. A. Simon în anul 1956, în scopul verificării axiomelor și teoremelor. Folosind cinci axiome și raționamentul invers, programul parcurgea, prin reducere, etapele de la teoremele de stabilit spre axiomele și teoremele date. Rezolvitorul de probleme generale, conceput în anul 1957 de aceiași cercetători, își propunea realizarea unei mașini destinate rezolvării unor probleme ce solicitau inteligență, precum și elaborarea unei teorii asupra modului de soluționare a acestor probleme de către oameni. Cele mai importante structuri de date cerute de program erau scopul și metodele sau procedurile de urmat.

*Mașina de verificare a teoremelor din geometrie* este un program scris în anul 1959 de H. Gelernter în limbaj FORTRAN extins — *FORTRAN List Processing Language* —, program implementat pe calculatoare de tipul IBM 704. Scopul programului era de a rezolva problemele complexe de geometrie plană.

Programul denumit *integrator simbolic automat* a fost lansat în anul 1961 la MIT. Scris în limbaj LISP și rulat pe calculatoare IBM 7090, programul era destinat rezolvării unora din primele probleme de inteligență artificială. În anul 1967 a apărut și un alt program de integrare simbolică — SINT (*S*imbolic *INT*egrator).

Programul STRIPS, lansat în anul 1971 de firma SRI International, era utilizat pentru orientarea unui robot într-un spațiu închis. O variantă dezvoltată după circa un an a permis implementarea capacității de a învăța.

În anul 1974 a apărut programul ABSTRIPS — o variantă modificată a programului STRIPS — folosit, de asemenea, pentru orientarea roboților într-un spațiu închis. Din compararea programelor menționate a rezultat că pentru rezolvarea unei probleme cu 11 operatori, pri-

mul avea nevoie de 30 minute, iar al doilea, de 5 minute și 28 secunde din timpul-calculator pentru a ajunge la soluția acceptată în scopul realizării unei deplasări complexe a robotului.

Exemplele menționate au confirmat cercetările teoretice întreprinse și au deschis un cîmp larg aplicațiilor de inteligență artificială.

## 1.4. REPREZENTAREA CUNOȘTINȚELOR

Natura cunoașterii și a inteligenței a fost analizată de psihologi, filozofi, lingviști, educatori și sociologi timp de sute de ani. Cercetările actuale sînt orientate spre tehnicile de reprezentare a cunoștințelor, prin care se înțeleg combinațiile de structuri de date și proceduri de interpretare utilizabile în programe în scopul evidențierii comportării inteligente și generării de inferențe. Reprezentarea cunoștințelor este în prezent cea mai dinamică ramură a inteligenței artificiale.

Cunoștințele folosite în reprezentări ale sistemelor de inteligență artificială sînt constituite din obiective, evenimente, performanțe și metacunoștințe — cunoștințe despre ceea ce știm. Utilizarea cunoștințelor în programe pentru recunoașterea obiectelor, aflarea răspunsului la întrebări sau manipularea roboților implică achiziționarea de noi date regăsirea faptelor din baza de cunoștințe și raționamentul ca o cale de găsire a soluției.

Prin achiziționarea cunoștințelor înțelegem ceva nou, cu semnificații deosebite, adăugat la ceva deja cunoscut. În domeniul sistemelor de inteligență artificială datele noi sînt clasificate de la început pentru a putea fi apoi regăsite ușor, de multe ori procesul de acumulare fiind însoțit de interacțiunea structurilor respective (fig. 1.5). Formele de reprezentare a cunoștințelor urmăresc apropierea de modul de lucru natural al omului. Fără interacțiunea structurilor de date nu poate fi îmbunătățită comportarea inteligentă a sistemelor.

Regăsirea faptelor din baza de cunoștințe folosește ca elemente de bază legăturile — în cazul existenței unor



determinări între o structură de date și alta, avînd ca rezultat un raționament — și concentrările — atunci cînd mai multe structuri de date sînt utilizate împreună pentru a forma un ansamblu mai larg.

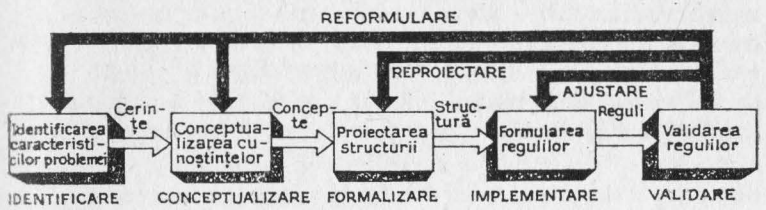



*Fig. 1.5.* Etapele achiziționării cunoștințelor.

Raționamentul este folosit de sistemele care trebuie să găsească răspunsuri la întrebările fără un corespondent direct în baza de cunoștințe. Există mai multe tipuri de raționament : formal, procedural, prin analogie, generalizare, abstractizare și prin metanivele.

Diferitele metode de reprezentare disponibile nu au condus încă la un răspuns clar asupra gradului de apropiere de memoria umană, cercetătorii din domeniul sistemelor de inteligență artificială nereușind să se pună de acord asupra unei teorii unitare.

Crearea de programe care să prezinte o comportare inteligentă se înscrie în cadrul epistemologiei experimentale. Specialiștii în inteligență artificială s-au implicat în găsirea tehnicilor adecvate pentru reprezentarea cunoștințelor și manipularea structurilor specializate de date care să conducă la inferențe. Cele mai cunoscute metode de reprezentare sînt de tip logic, procedural, rețele semantice, sisteme de producție și cadrele sau scenariile.

Logica formală a fost dezvoltată de filozofi, logicieni și matematicieni, pornind de la procesul de realizare a inferențelor din fapte. Cea mai importantă caracteristică a sistemelor logice o reprezintă nivelul înalt al corectitudinii deducțiilor, imposibil de obținut cu celelalte metode de reprezentare. Pentru logică sînt importante,

pe de o parte, axiomele sistemului sau relațiile și implicațiile care pot fi formalizate, iar pe de altă parte regulile de inferență determinante pentru structura deductivă rezultată atunci cînd se consideră adevărate anumite axiome.

Logica urmărește în primul rînd sinteza și determinarea noțiunii de adevăr care este fundamentală. Forma cea mai simplă este calculul prepozițional, care prin extindere se dezvoltă în calculul predicativ. Un predicat se aplică de obicei unui număr specific de argumente și are valoarea *adevărat* sau *fals* atunci cînd argumentele sînt obiecte specifice ; exemple de predicate sînt *mai-mare-decît, este-verde* etc. Două elemente suplimentare în logică sînt funcțiile și predicatul *egal-cu.*

De mare importanță în sistemele de inteligență artificială este logica de ordinul întîi în care se permite cuantificarea obiectelor specifice, dar nu și a predicatelor și funcțiilor, ceea ce determină claritate și completitudine. Exprimarea unei probleme de rezolvat rămîne însă partea cea mai importantă a logicii formale, pentru aceasta existînd mai multe metode de demonstrare automată a teoremelor.

Sistemele de inteligență artificială mai cunoscute care utilizează logica pentru reprezentarea cunoștințelor sînt QA3 (1969) — un sistem de întrebări și răspunsuri *(question-answering)* —, STRIPS (1972) — rezolvitorul de probleme al Institutului de Cercetări Stanford *(Stanford Research Institute Problem Solver)* destinat planificării mișcărilor unui robot pentru orientarea în spații închise — și FOL (1976) — utilizat ca mijloc interactiv sofisticat pentru studiul cu ajutorul logicii a problemelor epistemologice.

Reprezentările procedurale au condus la sisteme de inteligență artificială care știu să utilizeze cunoștințele, să găsească fapte semnificative și să realizeze inferențe. Printre primele sisteme se numără SIR (1968), folosit pentru întrebări și răspunsuri, PLANNER (1972) și GOLUX (1973). PLANNER este un limbaj de programare



care a condus la dezvoltarea unei familii întregi, incluzînd CONNIVER (1972), QA4 (1972), POPLER (1972) și QLIPS (1976).

Rețelele semantice din sistemele mai noi de inteligență artificială constau din noduri — reprezentate de obiecte, concepte și situații —, precum și din arcuri care definesc relațiile dintre ele. Sistemele din această grupă au evoluat de la *Teachable Language Comprehender* (1969), HAM (1973), *Active Structural Network* (1975) pînă la sistemele de înțelegere a vorbirii — TORUS (1975) — și la cele de deducție — SNIFFER (1977).

Sistemele de producție, bazate pe descrierea unor reguli care provin din perechea condiție-acțiune, au fost descrise încă din anul 1943. Un sistem de producție constă dintr-o bază de reguli, o structură de date-context și un interpretor care controlează toată activitatea.

Principalele domenii de aplicabilitate a sistemelor de producție sînt cele în care cunoștințele sînt constituite dintr-o multitudine de fapte (medicină), cele în care procesele pot fi reprezentate ca un set independent de acțiuni (monitorizarea pacienților), precum și cele în care cunoștințele pot fi separate ușor din ansamblu (biologie). Dintre programele dezvoltate se menționează : sistemul de producție adaptiv (1970), MYCIN (1976) — cu subsistemul de achiziție a cunoștințelor, TEIRESIAS —, AM, DENDRAL, HEARSAY (1975) și PROSPECTOR (1978).

DENDRAL, MYCIN și PROSPECTOR sînt de fapt sisteme-expert bazate pe cunoștințe și dezvoltate prin ingineria cunoașterii, care se referă nu numai la performanțele la niveluri înalte, ci și la transferul interactiv al expertizei — acumularea de cunoștințe de la experții umani și explicarea raționamentului utilizatorilor (fig. 1.6).

Modelele, diagramele, hărțile etc. s-au constituit în tr-o clasă de metode de reprezentare directă sau analogică. Cel mai important aspect privește corespondența dintre relațiile din structura de date a reprezentării și relațiile situației înfățișate. De exemplu, lungimea unui

segment pe hartă și lungimea reală sînt în corespondență directă. Avantajul principal al metodei reprezentării directe este dat de simplitatea actualizării reprezentărilor pentru a reflecta schimbările survenite în mediul exterior.

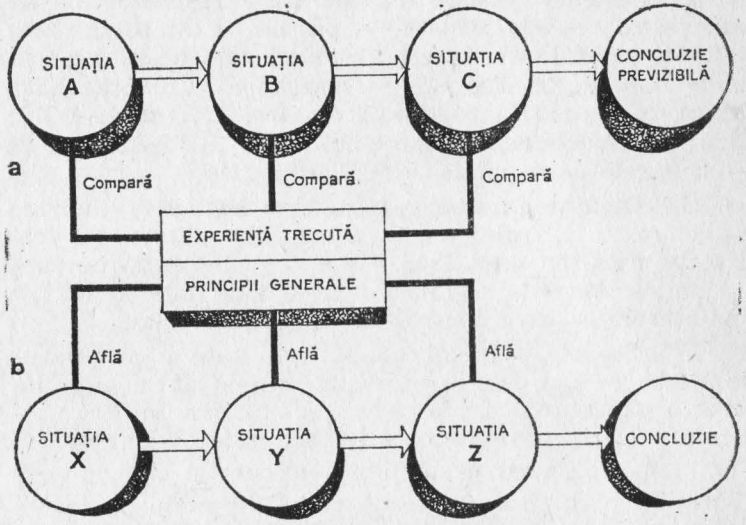



*Fig. 1.6.* Rezolvarea problemelor de către experți : a—situații cunoscute ; b—situații noi.

Reprezentarea analogică este inclusă în programe, ca de exemplu, sistemul pentru demonstrarea teoremelor din geometrie *(Geometry Theorem Prover, 1963)*, planificatorul general al spațiului *(General Space Planner,* 1976), WHISPER (1976) etc.

Sistemul pentru demonstrarea teoremelor din geometrie a fost unul dintre primele programe automate destinate verificării teoremelor prin tehnica reducerii problemelor și raționament invers. Planificatorul general al spațiului avea sarcina ordonării lucrurilor, pornindu-se de la unele constrîngeri. WHISPER este un sistem destinat efectuării unor raționamente numai prin reprezentări analogice.



Lucrările referitoare la primitivele semantice s-au concentrat asupra stabilirii vocabularului pentru reprezentări — problemă generală a tehnicilor analizate mai sus. Prin primitive semantice înțelegem simbolurile utilizate odată cu sistemul sau, în sens mai restrîns, elementele componente în care poate fi divizat înțelesul cuvintelor și propozițiilor.

Cadrele și scenariile sînt utilizate în reprezentarea cunoștințelor despre obiective și evenimente tipice ale unor situații specifice. Prin cadre pot fi înțelese percepțiile vizuale, dialogurile în limbaje naturale etc. ; scenariile sînt structuri similare cadrelor și sînt proiectate special pentru reprezentarea secvențelor din evenimente, ambele constituind de fapt metode de organizare a reprezentării cunoștințelor pentru simplificarea acestora și efectuarea de inferențe.

Pînă în prezent au fost elaborate mai multe sisteme practice — GUS (1977), limbajul de reprezentare a cunoștințelor (KRL — *K*nowledge *R*epresentation *L*anguage 1977), NUDGE (1977), KLONE (1978), UNITS (1980).

Din analiza efectuată pînă aici rezultă marea diversitate a metodelor și tehnicilor de reprezentare a cunoștințelor în sistemele de inteligență artificială.

Reprezentarea cunoștințelor este un capitol esențial pentru întregul domeniu al inteligenței artificiale, în care psihologia cognitivă joacă un rol deosebit de important. Încercările întreprinse se apropie mai mult sau mai puțin de operațiile presupuse a avea loc în creierul uman. Indiferent de domeniul asupra căruia se îndreaptă atenția, de la șah și matematică la chimie ș.a., scopul cercetărilor de reprezentare a cunoștințelor este de a permite programelor de inteligență artificială să se comporte ca și cînd ar ști ceva despre problema de rezolvat.

## 1.5. LIMBAJE DE PROGRAMARE

Cercetările în domeniul inteligenței artificiale se desfășoară în direcția elaborării de programe care să reliefeze o comportare inteligentă. Pentru implementarea acestei idei sînt necesare calculatoare electronice și lim-

baje de programare adecvate. Limbajele de programare asigură specificarea obiectelor și procedurilor necesare pentru rezolvarea diferitelor probleme. Principalele funcții ale limbajelor de nivel înalt sînt implementarea și modificarea programelor care demonstrează și testează ideile inteligenței artificiale și capacitatea de a gîndi.

Unul dintre primele limbaje de programare, IPL (1957), a introdus ideile de manipulare arbitrară a simbolurilor cu ajutorul calculatorului, precum și tehnica prelucrării listelor *(list processing)*. Inițial, majoritatea programelor de inteligență artificială — Chess, Logic Theorist, General Problem Solver, SAD-SAM, EPAM etc. — au fost scrise în limbaj IPL.

Prelucrarea listelor, precum și alte cîteva îmbunătățiri aduse modalităților de programare au stat la baza elaborării limbajului LISP (1958), care a devenit unul dintre cele mai utilizate sisteme din domeniul inteligenței artificiale. Structura listelor reprezintă un tip de date de nivel înalt, iar LISP oferă primitive pentru manipularea listelor și facilități de rulare care vin în ajutorul programatorului.

În ceea ce privește rezolvarea problemelor, a fost realizat și dezvoltat limbajul PLANNER (1971), care face apel la cunoștințe de gîndire. Programatorul exprimă instrucțiunile printr-o colecție de declarații, numite teoreme, pentru a descrie modul de atingere a stării finale, precum și ceea ce este de făcut în cazul apariției unor anumite situații pe parcurs. PLANNER permite codificarea cunoștințelor de procedură — adică modalitatea de realizare a acțiunilor — și oferă posibilitatea utilizării acestora în diferite moduri pentru rezolvarea problemelor.

O parte din limbajul PLANNER a fost implementată în MICROPLANNER, care cuprinde invocarea directă a modelelor procedurilor și raționamentul invers, automat. Există două tipuri de proceduri — teoreme-consecință și respectiv, antecedent. Ca exemplu de program scris în MICROPLANNER se menționează SHRDLU.

Limbajul CONNIVER (1972) a apărut ca reacție la rigiditatea concepției raționamentului în MICROPLANNER. Limbajul a introdus ideea de arbore al contextului



bazei de date, care reprezintă simultan diferite situații pentru diferite variante de acțiuni.

Limbajele PLANNER și CONNIVER au fost utilizate atît pentru reprezentarea cunoștințelor, cît și pentru programare ; ele aveau ca element comun important faptul că realizau structuri de date și control de înalt nivel, ușor de folosit. Ambele limbaje au deținut un rol deosebit în domeniul AI, dar în prezent nu mai sînt utilizate.

Pe lîngă LISP, PLANNER și CONNIVER, printre limbajele importante de programare se numără SAIL, QLISP, POP-2 și FUZZY.

Limbajul SAIL a fost lansat de Laboratorul de inteligență artificială de la Stanford (1969) ; el se bazează pe limbajul ALGOL-60, dar încorporează un formalism asociativ de refacere, denumit LEAP, care permite o examinare rapidă a conținutului bazei de date. Acest limbaj este destinat sistemelor de analiză a imaginilor și de înțelegere a vorbirii care implică efectuarea rapidă de calcule și utilizarea simbolurilor, ca în LISP.

QLISP este derivat din limbajul de rezolvare a problemelor, QA, apărut la mijlocul anilor '60. Există mai multe variante ale limbajului QLISP, dintre care menționăm QA1 pentru formalizarea ideilor din programul SIR, QA2 destinat aplicării mecanismului de efectuare a inferențelor în cazul demonstrării teoremei rezoluției, QA3 apărut ca o îmbunătățire a lui QA2 în ceea ce privește programul de sinteză, verificare și rezolvare a problemelor și QA4 (1971) folosit pentru eliminarea dificultăților referitoare la examinarea teoremelor și faptelor semnificative din baza de date. În acest scop este necesară precizarea cunoștințelor specifice domeniului, precum și a celor procedurale privind faptele utilizate, găsindu-se o formulă similară cu cea a limbajului PLANNER — indexarea în funcție de scopul urmărit. QA4 a fost primul limbaj în care diferitele afirmații se puteau reprezenta în mod unic pentru asocierea acestora cu proprietățile date. Mecanismul adoptat pentru context și structura generală de control sînt similare procedeelor folosite de limbajul CONNIVER.

Pentru a beneficia pe scară largă de avantajele prezentate mai sus, limbajul QA4 a fost implementat în

INTERLISP, formulînd limbajul QLISP. Spre deosebire de MICROPLANNER și CONNIVER, care sînt interpretatoare LISP, limbajul QLISP este un pachet de subrutine pentru INTERLISP, ambele fiind mai ușor de utilizat în programare. Ca și în cazul limbajului PLANNER, QLISP determină adevărul unei afirmații printr-o simplă examinare a bazei de date sau îl găsește prin deducții și teoreme logice. De asemenea se folosește noțiunea de set de proceduri. Dintre sistemele scrise în QLISP se menționează NOAH și DEDALUS.

Limbajul POP-2 (1967), similar cu ALGOL în ceea ce privește sinteza, are și multe dintre proprietățile limbajului LISP. POPLER este un limbaj în POP-2, încorporat în PLANNER.

FUZZY (1977) este un limbaj de programare elaborat pentru domeniul inteligenței artificiale. Printre sistemele care folosesc limbajul FUZZY se numără MYCIN, AIMDS/BELIEVER, HAM-RPM etc.

Dintre limbajele bazate pe calculul predicatelor de primul ordin se menționează PROLOG și FOL. Programele în limbaj PROLOG cuprind axiome în logica de primul ordin și o teoremă care trebuie demonstrată. Dacă teorema conține variabile cuantificate sistemul va returna imediat răspunsul care verifică teorema din punctul de vedere al adevărului, utilizînd metode similare cu cele ale limbajului QA3 ; chiar și modul de programare se aseamănă cu cel al limbajelor QA3 și PLANNER. Raționamentul invers folosit este automat, dar programatorul trebuie să adauge unele comenzi pentru a controla ordinea în care sînt luate în considerare propozițiile și axiomele. Pentru limbajul PROLOG a fost implementat un compilator care permite rularea programelor din diverse domenii în perioade de timp egale. Celălalt limbaj, FOL, avînd la bază metateoria pentru verificări, acceptă date logice și comenzi care pot fi elaborate și testate din punctul de vedere al corectitudinii. Structura de manipulare a teoriei permite construirea de metateorii arbitrare. În acest fel o teoremă poate fi demonstrată nu numai prin teorie, ci și prin metateorie. Demonstrarea unei teoreme prin metateorie corespunde executării unei proceduri pentru a produce o teoremă.



Marea varietate de aplicații a condus la elaborarea unui număr impresionant de limbaje și programe pentru sistemele de inteligență artificială. În cele ce urmează se prezintă principalele realizări în acest domeniu, cu precizarea că sistemele descrise sînt în majoritate specializate, proiectarea lor avînd în vedere rezolvarea unor probleme specifice.

Dezvoltarea mijloacelor pentru realizarea sistemelor de inteligență artificială implică o evoluție de tipul experimentare — testare — utilizare. Limbajele de programare actuale au apărut ca sisteme complexe, avînd caracteristici sofisticate, spre deosebire de limbajele de ingineria cunoașterii și programele de realizare a sistemelor care sînt limitate. În tabelul 1.4 sînt reprezentate principalele limbaje și sisteme hardware în domeniul inteligenței artificiale, avîndu-se în vedere categoriile, modul de implementare, destinația și descrierea limbajelor, precum și a echipamentelor care le folosesc în prezent.

TABELUL 1.4. **Limbaje și sisteme hardware în domeniul inteligenței artificiale**

| Modul de implementare | Destinația | Descrierea | Sistemele hardware |
|---|---|---|---|
| 1 | 2 | 3 | 4 |
| *Limbaje de programare* | | | |
| INTERLISP-D | uz general | proceduri | Xerox 1100 |
| LISP | uz general | proceduri | Mașini Lambda |
| PROLOG | uz general | logică/proceduri | DEC 10, DEC 20 |
| SMALLTALK-80 | uz general | obiecte | Tektronix 4404 |
| ZETALISP | uz general | proceduri | Symbolics 3600 |
| *Programe pentru realizarea sistemelor expert* | | | |
| EXPERT-EASE | achiziția cunoștințelor | inferențe | IBM-PC, IBM-XT, Victor 9000, DEC Rainbow |
| PLUME | interfețe pentru limbaje naturale | software | Symbolics 3600 VAX-11 |
| RULE-MASTER | achiziția cunoștințelor | inferențe | VAX-11, SUN, APOLLO |
| TIMM | achiziția cunoștințelor | inferențe | VAX-11/780, Prime 400 |

| 1 | 2 | 3 | 4 |
|---|---|---|---|
| *Limbaje de ingineria cunoașterii* | | | |
| ART | uz general | reguli/cadre/ procedură | Mașini CADR, Symbolics 3600 |
| DUCK | uz general | logică/reguli | Symbolics 3600, DEC VAX |
| KEE | uz general | reguli/cadre/ proceduri/ obiecte | Xerox 1100, Symbolics 3600 |
| KES | uz general | reguli/cadre | DEC VAX (sisteme de operare UNIX sau VMS) |
| M. 1 | uz general | reguli | IBM PC |
| OPS 5 | uz general | reguli | VAX-11 |
| OPS 5 e | uz general | reguli | Symbolics 3600 |
| OPS 83 | uz general | reguli/proceduri | VAX-11 |
| PERSONAL CONSULTANT | diagnoză | reguli | TI Professional Computer |
| S. 1 | uz general | reguli/cadre/proceduri | Xerox 1100, Xerox 1108 |
| SERIS | diagnoză | reguli | IBM PC |
| SRL + | uz general | cadre | Symbolics 3600, VAX-11 |

Programele pentru realizarea sistemelor actuale, metodele de reprezentare și limbajele utilizate de inteligența artificială sînt ilustrate în tabelul 1.5 ; în cazul limbajelor s-a indicat metoda de reprezentare.

TABELUL 1.5. **Programe pentru realizarea sistemelor, metode de reprezentare și limbaje utilizate de inteligența artificială**

| Denumirea | Categoria sistemului (L=Limbaj) | Denumirea | Categoria sistemului (L—Limbaj) |
|---|---|---|---|
| 1 | 2 | 1 | 2 |
| ACLS | Program | AL/X | L/Reguli |
| ADVISE | Program | AMORD | L/Reguli |
| AGE | Program | APES | L/Logică |
| AIMDS | L/Cadre | APLICOT | L/Logică |
| ALICE | L/Logică | ARBY | L/Reguli |



| 1 | 2 | 1 | 2 |
|---|---|---|---|
| ARS | L/Reguli | KRT | Program |
| ART | L/Reguli | KRYPTON | L/Cadre |
| C | L/Proceduri | LES | L/Reguli |
| LISP | L/Proceduri | LISP | L/Proceduri |
| CONCHE | Program | LOOPS | L/Obiecte |
| CSRL | L/Cadre | MARS | L/Reguli |
| DETEKTR | Program | MELD | L/Reguli |
| DPL | L/Cadre | MORE | Program |
| DUCK | L/Logică | MRS | L/Reguli |
| EMYCIN | L/Reguli | MUMPS | L/Proceduri |
| ERS | L/Reguli | M. 1 | L/Reguli |
| ETS | Program | NETL | L/Cadre |
| EXPERT | L/Reguli | OPS 5 | L/Reguli |
| EXPERT-EASE | Program | OPS 83 | L/Reguli |
| EXPERT-2 | L/Reguli | OWL | L/Cadre |
| EXPRS | L/Reguli | Personal consultant | L/Reguli |
| FIT | L/Logică | PICON | Program |
| FLAVORS | L/Obiecte | PIE | Program |
| FOL | L/Logică | PLUME | Program |
| FRL | L/Cadre | PRISM | L/Reguli |
| GEN-X | L/Reguli | PROLOG | L/Logică |
| GETREE | Program | PSL | L/Proceduri |
| GLIB | L/Reguli | PSYCO | L/Reguli |
| GPSI | L/Reguli | RADIAL | L/Reguli |
| GUESS/1 | L/Reguli | RITA | L/Reguli |
| HCPRVR | L/Logică | RLL | L/Cadre |
| HEARSAY-III | L/Reguli | ROGET | Program |
| HPRL | L/Cadre | ROSIE | L/Reguli |
| HSRL | L/Logică | ROSS | L/Obiecte |
| IMPULSE | L/Mijloc | RULE-MASTER | Program |
| IN-ATE/KE | L/Mijloc | RULE WRITER | Program |
| INTER-LISP | L/Proceduri | SAIL | L/Proceduri |
| INTER-LISP-D | L/Proceduri | SAVOIR | L/Reguli |
| KANDOR | L/Cadre | SEEK | Program |
| KAS | L/Reguli | SMALL-TALK | L/Obiecte |
| KBS | L/Obiecte | SOAR | L/Obiecte |
| KEE | L/Cadre | SRL | L/Reguli |
| KES | L/Reguli | SRL + | L/Cadre |
| KL-ONE | L/Cadre | STROBE | L/Cadre |
| KMS | L/Cadre | S. 1 | L/Obiecte |
| KRL | L/Cadre | | L/Reguli |

| 1 | 2 | 1 | 2 |
|---|---|---|---|
| TEIRESIAS | Program | Unit | |
| TIMM | Program | PACKAGE | L/Cadre |
| T. 1 | Program | ZETALISP | L/Proce-duri |

### 1.5.1. PROGRAME PENTRU REALIZAREA SISTEMELOR DE INTELIGENȚĂ ARTIFICIALĂ

Sistemele dezvoltate în etapa actuală implică una sau mai multe metode de reprezentare a cunoștințelor și programe de organizare. Principalele metode sînt ilustrate în fig. 1.7, în care sînt prezentate, de asemenea, programele utilizate și descrierea lor. S-au avut în vedere cele mai cunoscute metode pentru reprezentarea cunoștințelor — reguli, cadre și rețele —, deși unele aplicații folosesc combinații de tipul cadre-reguli.

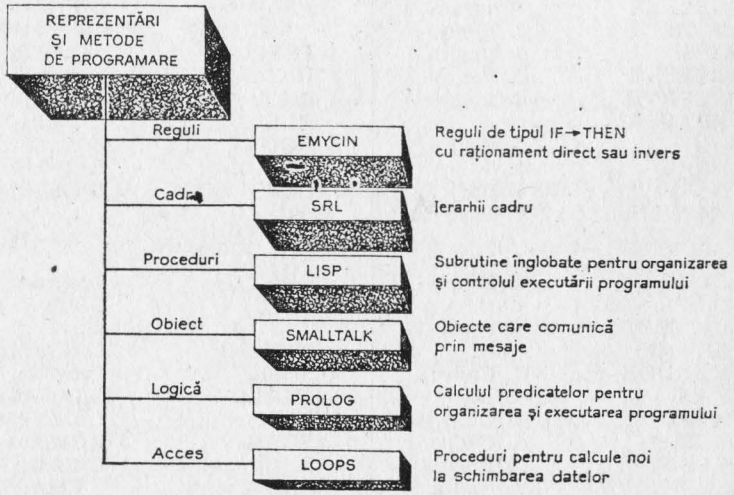



*Fig. 1.7.* Reprezentări și metode de programare pentru sisteme-expert.



Metodele bazate pe proceduri implică o cale normală de organizare a programului — subrutine înglobate în alte subrutine. Programatorul definește un set de proceduri de înalt nivel, care reprezintă de fapt un limbaj strict specializat pentru descrierea acțiunilor de urmat. În acest fel se asigură o mare flexibilitate a sistemului, dar explicitarea programelor este ceva mai dificilă decît în varianta standard.

Un mod special de organizare se realizează cu metodele bazate pe obiecte ; în acest caz entitățile capabile să prezinte o anumită comportare sînt reprezentate prin obiecte.

Metodele bazate pe logică folosesc programarea prin predicate pentru a controla analiza unui set de clauze declarative. Un program logic compară scopul dorit cu predicatul-consecință — definit ca fiind adevărat dacă predicatele, numite antecedent, sînt adevărate. Această metodă este similară cu raționamentul invers aplicat în cazul limbajelor bazate pe reguli.

Construirea unor afișaje vizuale sofisticate pentru variabilele monitorizate din program se face cu metode de acces. La schimbarea sau citirea datelor sînt folosite proceduri pentru monitorizarea programelor sau a dispozitivelor externe.

Principalele programe pentru realizarea sistemelor de inteligență artificială (fig. 1.8) sînt prezentate în sinteză pentru a se da o imagine asupra naturii și marii lor varietăți.

• *ACLS* (*A*nalog *C*oncept *L*earning *S*ystem) induce reguli din variantele de decizii luate de un expert uman în domeniul analizat. Utilizatorul specifică setul de atribute care determină decizia și, eventual categoria deciziilor care pot fi luate. Din exemplele cunoscute sistemul generează reguli de clasificare — atît sub forma unor arbori, cît și ca programe PASCAL. Metoda de inducție folosește algoritmul de învățare ID3. Sistemul descris a fost implementat pe microcalculatoare APPLE II+ și IBM PC.

• *ADVICE* constă dintr-un set integrat de sisteme de dezvoltare și are aplicații de uz general, incluzînd diverse forme de reprezentare a cunoștințelor, scheme de

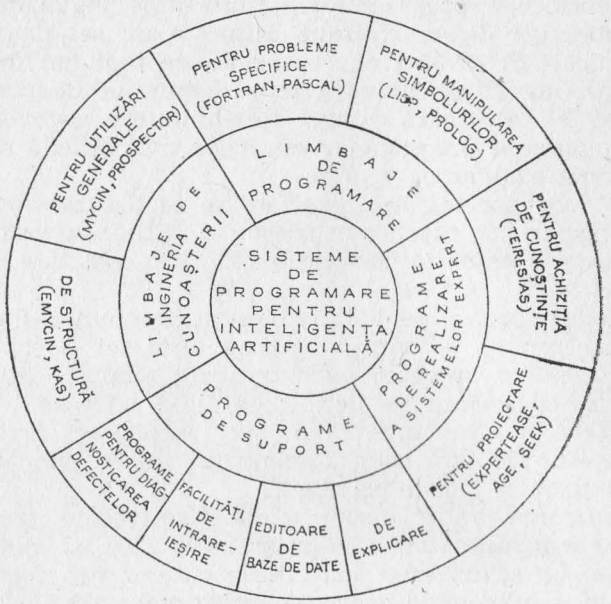

SISTEME DE PROGRAMARE PENTRU INTELIGENȚA ARTIFICIALĂ
LIMBAJE DE PROGRAMARE
LIMBAJE DE INGINERIA CUNOAȘTERII
PROGRAME DE REALIZARE A SISTEMELOR EXPERT
PROGRAME DE SUPORT
PENTRU PROBLEME SPECIFICE (FORTRAN, PASCAL)
PENTRU MANIPULAREA SIMBOLURILOR (LISP, PROLOG)
PENTRU ACHIZIȚIA DE CUNOȘTINȚE (TEIRESIAS)
PENTRU PROIECTARE (EXPERTEASE, AGE, SEEK)
DE EXPLICARE
EDITOARE DE BAZE DE DATE
FACILITĂȚI DE INTRARE-IEȘIRE
PROGRAME PENTRU DIAGNOSTICAREA DEFECTELOR
DE STRUCTURĂ (EMYCIN, KAS)
PENTRU UTILIZĂRI GENERALE (MYCIN, PROSPECTOR)

*Fig. 1.8.* Sisteme de programare pentru inteligența artificială.

propagare, strategii de control și programele de învățare inductivă GEM și CLUSTER. Sistemul este implementat în limbaj PASCAL și poate fi executat pe calculatoare DEC VAX în sistemul de operare UNIX.

• *AGE (A*ttempt to *GE*neralize) sprijină utilizatorul în selectarea cadrului de lucru, proiectarea unui limbaj de reguli și asamblarea părților într-un sistem-expert. *AGE* este constituit dintr-un set de blocuri, după modelul funcțiilor INTERLISP, cu aplicații în arhitecturi diverse de tipul mașinilor pentru inferențe. Sistemul este implementat în limbajul INTERLISP.

• *CONCHE (CON*sistency *CHE*cker) ajută la testarea completitudinii cunoștințelor mecanismelor, care au loc în reacțiile organice. După achiziționarea cunoștințelor, în urma unui dialog interactiv cu un expert chimist, programul extrage o serie de reguli. Ca și MYCIN, *CONCHE* folosește raționamentul invers și același mod



de derivare a regulilor. Rezultatele sînt prezentate sub forma regulilor, faptelor și condițiilor — realizate sau nerealizate. Implementarea s-a efectuat în limbajele LISP și FUZZY.

• *DETEKTR* (*D*evelopment *E*nvironment for *TEK*tronix *TR*oubleshooters) permite experților să elaboreze sisteme pentru efectuarea depanărilor și conține subsisteme specializate (diagrame, comunicații, reguli etc.) proiectate în limbajul GLIB. Programul are posibilitatea indicării reprezentărilor prin diagrame monitorizate. Implementarea s-a realizat în SMALLTALK-80 pentru sistemul de inteligență artificială Tektronix 4404.

• *ETS (E*xpertise *T*ransfer *S*ystem) este folosit în cercetare pentru analiza bazelor de cunoștințe scrise în limbajul OPS 5 prin intervievarea unui expert uman și producerea automată a setului de reguli, cu un factor de siguranță care poate fi revăzut și modificat pe parcurs. ETS este scris în limbaj INTERLISP-D și poate fi executat pe stații de lucru Xerox 1100.

• *EXPERT-EASE* este un program destinat construirii sistemelor-expert, pe baza unor caracteristici și factori asupra cărora este interogat specialistul. În acest fel sistemul poate învăța o procedură pentru rezolvarea problemei și produce un arbore decizional care reprezintă procedura. Programul este realizat în limbajul PASCAL și poate fi rulat pe calculatoare personale IBM PC sau IBM XT.

• *GETREE* conduce baze de date organizate în rețele, cu reguli reprezentate prin grafuri ȘI/SAU. Programul poate fi executat pe calculatoare DEC VAX în sistemul de operare VMS ; se mai utilizează o interfață grafică interactivă cu terminal alfanumeric VT 100, posedînd un set de caractere grafice, precum și un sintetizor de vorbire la ieșire.

• *GPSI (G*eneral *P*urpose *S*ystem for *I*nferencing) permite construcția sistemelor bazate pe reguli și constă dintr-un monitor pentru controlul global interactiv, o unitate de achiziție a cunoștințelor, o mașină de inferențe, un compilator de reguli și o interfață pentru utilizator. Mecanismele folosite conduc la o siguranță ridi-

cată și posibilitatea rulării sistemului-expert prin aplicarea mașinii de inferențe regulilor compilate.

• *IMPULSE* are facilități de editare organizate pe patru niveluri, pentru limbajul STROBE, fiecare cu editor și lista comenzilor asociate. Implementarea este realizată în INTERLISP-D pentru stații de lucru din seria Xerox 1100.

• *IN-ATE/KE* (*IN*telligent-*A*utomatic *T*est *E*quipment/*K*nowledge *E*ngineering) este destinat realizării sistemelor-expert pentru diagnosticarea și depanarea defectelor. Rețeaua semantică bazată pe reguli și baze de date, care conține informații asupra defectelor posibile și a punctelor de măsură, este similară cu cea a sistemului-expert PROSPECTOR din domeniul geologiei.

• *KRL* (*K*nowledge *R*epresentation *L*anguage) asigură corelarea rezultatelor obținute de grupe mari de specialiști. În acest fel se poate descrie modul de lucru al unui sistem de inginerie, organizarea și legătura dintre părți. Programul este scris în limbajele ZETALISP și FLAVORS pentru calculatoare de tipul Symbolics 3600.

• *MORE* generează reguli de diagnosticare prin intervievarea experților și solicitarea de informații suplimentare pentru modelarea domeniului. După achiziționarea cunoștințelor, el caută locurile înguste ale structurii și enunță unele sugestii asupra modului de îmbunătățire a regulilor. De asemenea sistemul verifică modul de alocare de către experți a factorilor de siguranță pentru reguli.

• *PICON* (*P*rocess *I*nteligent *CON*trol system) folosește metode de reprezentare bazate pe obiecte, cadre și reguli, precum și schemele de control prin raționament direct și invers. Achiziția cunoștințelor se face cu ajutorul unei interfețe grafice destinată transferului informațiilor structurale, descrierii procesului și euristicii în baza de cunoștințe. Programul asigură explicarea și actualizarea dinamică a expertizelor sistemului. Ca limbaje de programare se menționează ZETALISP și C pentru rularea pe stații de lucru de tip Lambda/Plus.

• *PIE* (*P*ersonal *I*nformation *E*nvironment) este un sistem experimental care permite extinderea posibilităților oferite de SMALLTALK-76, în principal a posibilității reprezentării și manipulării desenelor pe baza perspectivelor multiple.



• *PLUME* asigură interfața limbajelor naturale la sistemele-expert, folosind interpretorul DYPAR II. Prin extinderea lexicului și introducerea de cadre specifice și reguli gramaticale se poate extinde programul la diferite domenii. Sistemul este implementat în limbaj COMMON LISP și poate fi rulat pe sisteme DEC VAX, subsistemul de operare VMS și pe stațiile de lucru cu calculatoare Symbolics 3600 sau cu cele provenind de la Carnegie Group.

• *ROGET* — denumirea programului provine de la numele lexicografului englez Peter Mark *Roget* (1779—1869) — ajută la alcătuirea de către specialiști a bazei de date pentru sisteme-expert de diagnosticare. Modul de lucru este interactiv, programul formulînd o serie de întrebări pentru identificarea subproblemelor, evidențierea informațiilor necesare pentru găsirea soluțiilor și a relațiilor dintre faptele existente. Sistemul este scris într-o versiune extinsă a limbajului EMYCIN și are aplicații în medicină.

• *RULEMASTER* asistă dezvoltarea sistemelor-expert bazate pe reguli, folosind limbajul RADIAL pentru exprimarea regulilor și un sistem de inducție care produce reguli din exemple. Dintre celelalte caracteristici mai importante se menționează structurarea ierarhică a regulilor, acceptarea datelor externe și a prelucrării programelor concepute în sistemul de operare UNIX, generarea automată a explicațiilor și posibilitatea folosirii operatorilor și a tipurilor de date impuse de beneficiar. *RULEMASTER* este implementat în limbajul de programare C pe minicalculatoare și microcalculatoare VAX și stații de lucru Sun.

• *RULE WRITER* asigură formularea de către specialiștii în ingineria cunoașterii a regulilor în limbajul EXPERT. Mecanismul de lucru se bazează pe exemple simulate și conduce la generarea unor modele pentru procesul de inducție a regulilor. Programul este implementat în limbaj LISP.

• *SEEK* este un program de cercetare destinat ajustării și generalizării regulilor de dezvoltare a sistemelor

-expert pentru diagnosticare prin reprezentarea regulilor în limbajul EXPERT într-un format tabelar. Sistemul este interactiv, permițînd utilizatorului să decidă asupra modului de generalizare sau specializare a componentelor. Programul este scris în limbaj FORTRAN pentru calculatoare DEC-20.

• *TEIRESIAS* ajută la transferul interactiv de cunoștințe de la specialiști în baza de cunoștințe. Mecanismul de lucru asigură explicitări și verificări simple asupra consistenței bazei de cunoștințe. Implementarea s-a făcut în limbajul INTERLISP.

• *TIMM* (*T*he *I*ntelligent *M*achine *M*odel) asistă specialiștii în realizarea sistemelor-expert pe baza unei liste a tuturor deciziilor posibile, precum și a numelor și valorilor factorilor, elemente ce trebuie avute în vedere prin generarea unor inferențe de tipul regulilor IF-THEN. Programul acceptă informații probabilistice și este dezvoltat în limbajul FORTRAN pentru calculatoare VAX 11/780.

• *T. 1* este un pachet de programe pentru instruirea specialiștilor și factorilor de decizie asupra ingineriei cunoașterii. Prezentarea se face prin lecții înregistrate pe videocasete, exerciții de laborator, sisteme de demonstrare a software-ului etc. Conceptul descris utilizează o versiune modificată a limbajului M. 1, iar rularea se face pe calculatoare personale IBM PC în sistemul de operare PC DOS 2.0.

### 1.5.2. LIMBAJE BAZATE PE CADRE

• *AIMDS* este un limbaj de ingîneria cunoașterii pentru reprezentări prin cadre (frame) și proceduri. Principalele caracteristici sînt : raționamente deductive și nedeductive, detectarea automată a nepotrivirilor în baza de cunoștințe, utilizarea probabilităților de orientare a acțiunilor și efectuarea inferențelor analogice. *AIMDS* este implementat în FUZZY, care la rîndul său are la bază limbajul UCI-LISP.

• *CSRL* (*C*onceptual *S*tructures *R*epresentation *L*anguage) asigură reprezentarea conceptelor într-o ierarhie



de diagnostice. Programele de suport includ un verificator al sintaxei, comenzi pentru invocarea conceptelor cu mesaje etc. Acest program este scris în limbajul ELISP, folosește o versiune a sistemului FRL și poate fi rulat pe calculatoare electronice DEC 20/60.

• *DPL* (*D*esign *P*rocedure *L*anguage) creează și manipulează reprezentări bazate pe cadre ale proiectelor LSI. Într-un singur set de structuri de date sînt reprezentate uniform atît informațiile de structură, cît și cele de proiectare, realizîndu-se în acest fel o bază de date pentru a răspunde la întrebări privind modul de conectare a dispozitivelor sau care simulează comportarea dispozitivului. Programele-suport asigură un editor grafic interactiv pentru a construi proiecte LSI sub forma procedurilor DPL. Implementarea s-a făcut în ZETALISP, iar rularea pe calculatoare Symbolics 3600.

• *FRL* (*F*rame *R*epresentation *L*anguage) — care este un sistem experimental — include programe suport destinate succesiunilor multiple, constrîngerilor — cerințe și preferințe —, abstractizărilor, procedurilor atașabile etc.

• *HPRL* (*H*euristic *P*rogramming and *R*epresentation *L*anguage) integrează o schemă de reprezentare a cadrelor cu una bazată pe reguli. Executarea regulilor se face printr-un set de funcții LISP sub supervizarea unui interpretor specific. Programul utilizează raționamentul direct și invers și metareguli care permit beneficiarului să construiască programe complexe. El este o extensie a programului FRL implementat în PSL — un dialect al limbajul LISP — și poate fi executat pe sisteme de calculatoare electronice DEC VAX și HP-9835.

• *KANDOR* — un sistem experimental cu putere limitată ce asigură utilizatorului unele facilități — care are la dispoziție un set de interfețe simple. Programul este scris în limbaj LISP.

• *KEE* (*K*nowledge *E*ngineering *E*nvironment) este un sistem de uz general, dezvoltat inițial pentru aplicații de genetică moleculară. Pe lîngă reprezentări de cadre, programul mai admite reguli, proceduri și obiecte. Dintre caracteristicile principale se menționează : bazele mul-

tiple de cunoștințe pentru asigurarea proiectării într-un sistem modular și a raționamentului direct și invers al interpretorului de reguli. Programele-suport includ un pachet orientat grafic pentru diagnosticarea defectelor și facilități de explicare cu sistem de afișare a lanțului de inferențe. *KEE* este scris în limbaj LISP pentru calculatoare de tip Xerox 1100 și Symbolics 3600.

• *KL-ONE* este un limbaj de ingineria cunoașterii pentru reprezentări bazate pe cadre cu posibilități de realizare automată a succesiunilor, programe-suport pentru rețele semantice și un mod automat de clasificare. Ca suport se folosesc un editor grafic al bazei de date și programe de afișare. *KL-ONE* este implementat în limbaj INTERLISP-D pentru mașini LISP Xerox 1100 și sisteme de calculatoare electronice DEC VAX.

• *KMS* (*K*nowledge *M*anagement *S*ystem) este un sistem de cercetare care, pe lîngă cadre, acceptă și reguli, fiind alcătuit din subsisteme, fiecare cu modul său de reprezentare a cunoștințelor și metoda proprie de efectuare a inferențelor. Caracteristicile principale sînt : deducția pe bază de reguli, clasificarea după un model statistic, discriminant liniar etc. *KMS* este scris în limbajul LISP pentru calculatoare de tipul Univac 1100/40.

• *KRYPTON* este un limbaj de ingineria cunoașterii pentru reprezentări bazate pe cadre și pe metode logice, incluzînd o componentă care definește cadrele și rețelele și alta pentru menținerea unei baze de date de afirmații logice. Pentru programare este folosit limbajul INTERLISP-D.

• *NETL* este un sistem de cercetări în limbaj MACLISP pentru reprezentări prin cadre și rețele semantice, avînd posibilitatea de a crea și manipula copii virtuale complexe și de mari dimensiuni ale rețelelor.

• *OWL* este un limbaj care include cadre de rețele semantice, concepte de specializare și mecanism flexibil pentru succesiuni. Baza de date este unitară, conținînd toate cunoștințele și un set de programe în cod mașină și LISP. Implementarea sistemului de cercetare *OWL* este realizată în limbajul LISP.

• *RLL* (*R*epresentation *L*anguage *L*anguage) este un sistem flexibil de cercetare care permite beneficiarului să



specifice un limbaj caracteristic domeniului prin nominalizarea setului de reprezentări, elaborarea strategiei succesiunilor, precum și încorporarea structurilor generale LISP ca zone de cadre. Suportul este asigurat de un editor complex pentru verificarea sintaxei și semanticii. *RLL* este scris în limbajul INTERLISP.

• *SRL* (*S*chema *R*epresentation *L*anguage) — sistem de cercetare pentru reprezentări bazate pe cadre — include contexte multiple și relații de succesiune automată care pot fi specificate de utilizator. Programul asigură un set de primitive pentru definirea relațiilor și semanticii succesiunilor, precum și parametrii examinării pentru modificarea procedurii de examinare. Fiecare cadru, care în *SRL* devine o schemă, poate avea metacunoștințe asociate. Un dialect al acestui program, *SRL*/1.5, este implementat în limbajul FRANZ LISP pentru calculatoare electronice DEC VAX sub sistemul de operare UNIX.

• *SRL+* este un limbaj de ingineria cunoașterii pentru reprezentări bazate pe cadre, logică, reguli și obiecte. Caracteristicile principale sînt : posibilitatea definirii de către beneficiar a relațiilor de succesiune, atașarea procedurilor, simularea cu ajutorul unor termeni adecvați și manipularea erorilor printr-un mecanism specific. Programele-suport cuprind un sistem de management al bazei de date, grafică bidimensională și interfață pentru limbajul natural PLUME de analiză gramaticală, *SRL+* poate fi scris în limbajele COMMON LISP și FRANZ LIPS pentru stațiile de lucru ale grupului Carnegie, calculatoare DEC VAX sub sistemul de operare VMS și stații de lucru Symbolics 3600.

• *UNIT PACKAGE* este un limbaj de ingineria cunoașterii pentru reprezentări prin cadre, dezvoltat ca sistem de cercetare. Cadrele pot fi organizate în rețele semantice separate. Programul cuprinde suportul pentru generalizarea relațiilor în structuri ierarhice, compararea modelelor și un mecanism din proceduri atașate. *UNIT PACKAGE* este implementat în INTERLISP sub sistemele de operare TENEX și TOPS 20.

### 1.5.3. LIMBAJE BAZATE PE LOGICĂ

• *ALICE (A Language for Intelligent Combinatorial Exploration)* este un limbaj de ingineria cunoașterii pentru reprezentări bazate pe logică, utilizat ca sistem experimental. Caracteristicile principale sînt : vocabularul extins pentru descrierea problemelor combinatorii în cercetarea operațională și un proces de examinare a stărilor finale prin alegerea celor mai bune soluții și verificarea acestora.

• *APES (A Prolog Expert system Shell)* asigură o interfață interactivă pentru scrierea sistemelor-expert în limbajul de programare logică micro-PROLOG. Regulile din baza de date sînt exprimate în propoziții de tip PROLOG. Suportul este constituit din module definite de utilizator sau similare celor de tip MYCIN și bayesian. Sistemul comercial este implementat în limbaj micro-PROLOG pentru mai multe tipuri de minicalculatoare electronice, dintre care se menționează IBM PC.

• *APLICOT* — dezvoltat de experții japonezi — include o schemă flexibilă de control cu raționament direct și invers, manipulator al factorului de siguranță și mod explicativ. Suportul este constituit dintr-un pachet PROLOG și un editor interactiv. *APLICOT* este scris în limbaj PROLOG DEC-10 pentru sistemul de calculatoare ICOT DEC-2060.

• *DUCK* este un limbaj care combină patru modele ale inteligenței artificiale — programarea logică, sistemele bazate pe reguli, raționamentul nemonoton și examinarea deductivă —, folosind o notație din calculul predicatelor de ordinul întîi. Schemele de control sînt cu raționament direct și invers, implementarea fiind realizată în limbaj NISP, un dialect LISP, pentru calculatoare DEC VAX, subsistemele de operare UNIX și VMS și stațiile de lucru Symbolics 3600 și Apollo.

• *FIT* integrează caracteristicile relaționale din PROLOG și cele funcționale din LISP într-un nucleu și o interfață interactivă cu utilizatorul. Algoritmii sînt definiți ca funcții de relații, iar conjuncțiile logice sînt evaluate simultan prin paralelism ȘI. *FIT* este scris într-un set funcțional din UCI-LISP pentru calculatoare DEC-10.



• *HCPRVR (Horn Clause theorem PRoVeR)* este un interpretor și limbaj de programare bazat pe logică. Programele sînt constituite din liste de axiome ordonate, fiecare reprezentînd o formulă atomică sau o implicație. Interpretorul este un sistem de demonstrare a teoremelor, bazat pe propoziții de tip Horn. *HCPRVR* este scris în limbaj LISP pentru sisteme de calculatoare DEC PDP-10.

• *HSRL* — un limbaj de ingineria cunoașterii cu posibilități de programare logică — combină reprezentările logice și prin cadre în *SRL*. Suportul este constituit de un mod interactiv de creare, editare și executare de programe. Interpretorul este o versiune modificată a celui din programul logic HCPRVR. *HSRL* este conceput în limbajul FRANZ LISP ca extensie SRL în sistemul de operare UNIX.

• *PROLOG* este un limbaj de programare pentru reprezentări logice ; el cuprinde reguli sau legături de inferențe pentru demonstrarea relațiilor între obiecte. Interpretorul *PROLOG* caută să dovedească adevărul relațiilor specificate prin raționament invers. În prezent există multe variante de *PROLOG*, unele dintre ele utilizînd condiții din *LISP* pentru a-i asigura o flexibilitate ridicată. De asemnea există un număr impresionant de tipuri de calculatoare pe care pot fi rulate programe în limbaj *PROLOG*, dintre care se menționează DEC 10 și DEC 20.

### 1.5.4. LIMBAJE PENTRU REPREZENTĂRI PRIN OBIECTE

• *FLAVORS* include definiții simple de tip abstract, ierarhii generalizate și permite asocierea listelor cu obiectele. *FLAVORS* este un sistem în limbaj ZETALISP și poate fi rulat pe stații de lucru din seria Symbolics 3600.

• *KBS (Knowledge-Based Simulation system)* implică modele de simulare și metode de reprezentare prin obiecte și cadre. Simularea se face prin modele bazate pe obiecte. Suportul conține programe de construcție și modificare interactive și facilități pentru verificarea con-

sistenței și completitudinii. Experimental, *KBS* este implementat în *SRL* pentru sisteme VAX FRANZ LISP.

• *LOOPS* — un limbaj pentru reprezentări prin obiecte, reguli, acces și proceduri — integrează patru scheme de programare în vederea construirii sistemului. Suportul conține programe pentru întreruperi și editare. *LOOPS* este scris în limbaj INTERLISP-D pentru stații de lucru din seria Xerox 1100.

• *ROSS* (*R*ule-*O*riented *S*ystem for *S*imulation) permite realizarea succesiunii de atribute de la mai multe surse, precum și combinarea liberă a comenzilor cu apelările funcțiilor LISP. Programele de suport cuprind un editor pentru afișare și un pachet de abrevieri pentru asigurarea unei citiri rapide. *ROSS* este implementat în limbaj FRANZ LISP, ca sistem de cercetare, pentru a fi rulat pe calculatoare electronice DEC VAX subsistemul de operare UNIX.

• *SMALLTALK* este un limbaj de programare pentru reprezentări prin obiecte organizate ierarhic în clase. Suportul constă dintr-o interfață interactivă cu grafică de înaltă rezoluție și un dispozitiv de indicare. Sistemul a fost dezvoltat în diferite versiuni, începînd din anii '70, ca sistem de cercetare pentru stațiile de lucru Xerox 1100, ultima formă, *SMALLTLK-80*, putînd fi folosită pe stații de lucru tip TEK 4404.

• *STROBE* asigură dezvoltarea suportului pentru INTERLISP. Caracteristicile principale constau în : baze de cunoștințe cu adrese multiple, programe de suport pentru ierarhii generalizate, mecanism flexibil de succesiuni, invocare indirectă a procedurilor etc. Sistemul permite noduri non-obiect în baza de cunoștințe, cum ar fi expresiile *S*, funcțiile LISP, hărțile în biți și ariile. *STROBE* este implementat în limbajul INTERLISP.

### 1.5.5. LIMBAJE BAZATE PE PROCEDURI

• *C* este un limbaj de programare pentru reprezentări bazate pe proceduri. Nivelul și eficiența arătate au condus la utilizarea acestui limbaj, cunoscut și sub denumirea de limbaj de programare a sistemelor, pentru



scrierea sistemelor de operare. Sistemul *C* operează direct cu numere, caractere și adrese, spre deosebire de limbajele de nivel mai înalt care folosesc șiruri, seturi, liste sau arii. Proiectat să lucreze cu sistemul de operare UNIX, limbajul *C* poate fi rulat pe o mare varietate de calculatoare electronice, ca de exemplu DEC VAX.

• *COMMON LISP* constituie un nou dialect LISP, conceput pentru a face acest limbaj compatibil cu un număr mare de sisteme de calculatoare electronice ; el încorporează un set de tipuri de date complexe și o structură extinsă de control. Sistemul comercial este furnizat de diferite firme sub diferite denumiri — GCLISP, VAX LISP etc.

• *INTERLISP*, ca dialect LISP, are toate caracteristicile acestuia și în plus programe de suport pentru diagnosticarea defectelor, un editor și posibilitatea corectării rapide a multor erori. Sistemul poate fi rulat pe diferite tipuri de calculatoare, cum ar fi, de exemplu, mașinile Xerox.

• *INTERLISP-D* este un limbaj de programare pentru reprezentări prin proceduri, conceput ca o versiune de *INTERLISP*. Sistemul asigură toate caracteristicile din *INTERLISP*, precum și programe-suport suplimentare pentru monitoare grafice ; el poate fi rulat pe calculatoare electronice Xerox 1100.

• *LISP* (*LISt Processing* language) — un limbaj flexibil pentru reprezentări prin proceduri — asigură un număr redus de funcții primitive, pe baza cărora utilizatorul poate proiecta funcții de înalt nivel pentru cele mai diverse aplicații. Mecanismul de manipulare a simbolurilor sub forma structurilor de liste permite reprezentarea conceptelor complexe. *LISP* are un management automat al memoriei și tratează uniform codurile și datele, ceea ce conduce la posibilitatea modificării codurilor proprii. Sistemul descris este cel mai cunoscut limbaj de inteligență artificială și are un număr mare de dialecte — INTERLISP, MACLISP, FRANZ LISP, COMMON LISP și ZETALISP.

• *MUMPS*, ca și LISP, este un limbaj care permite variabile compuse și atomi. Spre deosebire de LISP, programele-suport pentru arii multidimensionale sînt re-

prezentate prin arbori cu descendenți ordonați după valorile de indexare. Ca urmare, este facilitată sortarea automată după anumite chei introduse în program. Inițial *MUMPS* a fost destinat efectuării calculelor în medicină, în prezent fiind disponibil pentru mai multe tipuri de sisteme de calculatoare electronice.

• *PSL (P*ortable *S*tandart *L*ISP) este un sistem de cercetare apărut ca dialect al limbajului LISP. Programele-suport cuprind un compilator integrat, un editor și facilități pentru diagnosticarea defectelor. Interpretorul este scris chiar în limbaj PSL, pentru sisteme de calculatoare electronice ca DEC System-20, DEC VAX și stații de lucru Apollo.

• *SAIL* este un limbaj de programare derivat din ALGOL 60, avînd memorie asociativă, un set larg de funcții pentru manipularea datelor și intrări/ieșiri de nivel redus. Sistemul mai cuprinde legături flexibile între proceduri în limbaje de asamblare, facilități de multiprocesare, macrocompilator etc. Programele-suport sînt de asemenea complexe și interactive cu utilizatorul. *SAIL* a fost proiectat pentru a lucra pe calculatoare PDP-10 sub sistemele de operare TOPS-10 și TENEX.

• *ZETALISP* este un dialect LISP, bazat pe MACLISP, cu o serie de extinderi și îmbunătățiri. Suportul include un dispozitiv de afișare de înaltă rezoluție, un editor de programe/texte integrat etc. *ZETALISP* poate fi rulat pe sisteme de calculatoare Symbolics LM-2 și 3600, precum și pe mașini de tip Lambda și Lambda/Plus

### 1.5.6. LIMBAJE BAZATE PE REGULI

• *AL/X (A*dvice *L*anguage X) este un limbaj de ingineria cunoașterii pentru reprezentări prin reguli cu o schemă de control care folosește raționamentul direct și invers. Programele-suport explică modul de efectuare a raționamentelor de către sistem. *AL/X* este implementat în limbaj PASCAL pentru calculatoare PDP 11/34 sub sistemul de operare UNIX.

• *AMORD (A M*iracle *O*f *R*are *D*evice) este un sistem de cercetare care cuprinde rețele de discriminare



pentru afirmații și reguli, o schemă de control cu raționament direct și un sistem de întreținere. Sistemul este scris în MACLISP.

• *ARBY* include notația folosită la calculul predicatelor pentru exprimarea regulilor, raționamentul direct și o interfață cu utilizatorul. Suportul conține facilități de editare a regulilor și a cadrelor de interacțiune. *ARBY* este implementat în limbaj FRANZ LISP și utilizează sistemul de refacere de uz general DUCK.

• *ARS (A*ntecedent *R*easoning *S*ystem) reprezintă regulile pentru rezolvarea problemelor în mod specific, o bază de date asociativă și raționament direct. Toate deducțiile făcute sînt păstrate pentru aplicații și eliminarea contradicțiilor. *ARS* este scris în MACLISP pentru sistemele de operare MULTICS, TOPS-10 și ITS.

• *ART (A*dvanced *R*easoning *T*ool) este un limbaj de ingineria cunoașterii pentru reprezentări prin reguli, cadre și proceduri. Schemele de control folosesc raționamentul direct și invers, iar baza de date este structurată prin definirea contextului în care apar faptele și regulile. *ART* este scris în limbaj LISP pentru mașini CADR și Symbolics 3600.

• *EMYCIN (E*ssential *MYCIN)* include o schemă de control cu raționament invers și facilități automate pentru integrare. Programele-suport asigură interfața pentru explicarea modului de gîndire al sistemului și pentru achiziția de noi cunoștințe. Implementarea este în limbaj INTERLISP pentru calculatoare DEC PDP-10, subsisteme de operare TENEX sau TOPS 20.

• *ERS (E*mbedded *R*ule-based *S*ystem) are o semantică similară cu PROSPECTOR, tehnici logice de tip Fuzzy și Bayes, interogare automată și posibilitatea apelării unui set de funcții primitive specifice aplicației pentru evaluarea nodurilor evidente. Suportul este constituit din programe pentru afișare grafică și pentru explicarea modului de gîndire al sistemului. *ERS* este scris în limbaj PASCAL pentru calculatoare IBM PC-XT și DEC VAX 11/780.

• *EXPERT* este un limbaj a cărui schemă de control folosește raționamentul direct pentru diagnosticare sau pentru soluționarea unor probleme de clasificare. Pro-

gramele-suport conțin interfețe cu utilizatorul sofisticate pentru explicarea, achiziția și verificarea consistenței. *EXPERT* este implementat în limbaj FORTRAN pentru echipamente DEC și IBM. Sistemul *EXPERT* este unul dintre cele mai utilizate limbaje pentru cercetare în medicină.

• *EXPERT-2* include un interpretor de reguli cu raționament invers și programe-suport care permit accesul complet la sistemul FORTH. Sistemul este implementat în limbaj FORTH pentru minicalculatoare de tipul Apple II.

• *EXPRS (EXpert PRolog System)* permite un format al regulilor asemănător limbii engleze și o schemă de reprezentare a cunoștințelor cu mecanism de inferențe folosind raționamentul direct și invers. Sistemul generează în mod automat răspunsuri la întrebări asupra tipului de reguli stabilite și cuprinde facilități de explicare. *EXPRS* este conceput în limbaj PROLOG ca sistem de cercetare pentru calculatoare DEC-10.

• *GEN-X (GENeric-eXpert system)* este un limbaj pentru reprezentări prin reguli și cadre. Sistemul cuprinde managementul cunoștințelor — destinat creării și editării bazei de cunoștințe —, interpretorul pentru diversele reprezentări și generatoare de cod care traduc baza de cunoștințe în programe scrise în limbajele C, ADA, PASCAL și FORTRAN. *GEN-X* este implementat în limbajul de programare C pentru IBM PC și alte tipuri de minicalculatoare.

• *GLIB (General Language for Instrument Benavior)* a fost dezvoltat pentru achiziția de reguli în depanarea instrumentelor electronice. Sistemul are un vocabular și o sintaxă specifice pentru exprimarea regulilor, datelor tehnice și înregistării comportării dispozitivelor analogice. *GLIB* este conceput în limbaj SMALLTALK-80 pentru sistemul de inteligență artificială Tektonix 4404.

• *GUESS/1 (General pUrpose Expert Systems Shell)* este un sistem de cercetare care permite reprezentarea prin reguli ,tabele relaționale, arbori ierarhici, rețele semantice și cadre. Cunoștințele de control sînt incluse în cadre și reguli care folosesc raționamentul direct și invers. Programele-suport includ facilități de explicare și limbaje naturale. *GUESS/1* este scris în PROLOG pentru a fi rulat pe calculatoare DEC VAX 11/780 sub sistemul de operare VMS.



• HEARSAY-III integrează regulile într-o arhitectură specifică, sursele de cunoștințe avînd posibilitatea să comunice printr-o bază de date centrală. Sistemul se aplică la proiectele care necesită prelucrarea asincronă a informațiilor și la cele care au o structură de control pentru manipularea unor stări finale multiple. *HEARSAY III* este scris în limbaj LISP ca un sistem de cercetare.

• *KAS (Knowledge Acquisition System)* este, de fapt, sistemul PROSPECTOR, din care lipsesc informațiile din domeniul geologiei. Sistemul folosește reguli pentru inferențe cu factori de siguranță asociați și o rețea de cunoștințe divizată pentru codificarea cunoștințelor. Inferențele se obțin prin raționament direct și invers, iar propagarea probabilității, prin rețea semantică. Programele-suport asigură facilități pentru explicarea și achiziționarea cunoștințelor, recunoșterea sinonimelor, revizuirea răspunsurilor. Implementarea s-a făcut în limbaj INTERLISP.

• *KES (Knowledge Engineering System)* conține o schemă de control cu raționament invers, mecanism de manipulare pentru siguranță și un sistem de clasificare după modele statistice. Suportul este constituit de interfețele pentru explicarea modului de gîndire și de achiziționare a noi cunoștințe. *KES* este scris în limbaj FRANZ LISP pentru a fi rulat pe sisteme VAX DEC și UNIVAC.

• *LES (Lockheed Expert System)* este un sistem de cercetare pentru reprezentarea prin reguli și cadre. Caracteristicile principale sînt : schemele de control cu raționament direct și invers, agenda pentru stări finale importante și facilitățile pentru limbaje naturale și explicare. *LES* este scris în limbajul PL/1.

• *MARS (Multiple Abstraction Rule-based System)* reprezintă prin reguli simulatoarele ierarhice, discrete, comandate de evenimente. Structura este specificată ie-

rarhic, raționamentul este direct și invers, iar simularea se face în mod simbolic. Programele-suport conțin un compilator pentru specificațiile bazate pe reguli și proceduri speciale pentru simulare. Părțile unui proiect pot fi simulate cu niveluri diferite de abstractizare. *MARS* este conceput în limbajul MRS ca un sistem experimental.

• *MELD (ME*ta-*L*evel *D*iagnosis*)* este un limbaj pentru reprezentarea prin reguli și cadre. Regulile cauzale și euristice de nivel obiect sau metanivel sînt separate. Raționamentul este direct și invers, iar strategia de selecție a regulilor de metanivel utilizează rezoluția de la OPS 5, în care este implementat sistemul.

• *MRS (M*etalevel *R*epresentation *S*ystem*)* este un sistem de cercetare pentru reprezentarea logică și prin reguli. Schema de control flexibilă permite folosirea raționamentului direct și invers, demonstrarea teoremei rezoluției și are posibilitatea reprezentării cunoștințelor de metanivel. *MRS* este scris în limbajul INTERLISP.

• *M.1* este implementat în PROLOG pentru calculatoare IBM PC sub sistemul de operare PC DOS 2.0. Schema de control folosește raționamentul invers, iar limbajul are o sintaxă similară limbii engleze. Suportul este asigurat prin programe de depanare grafice și facilități pentru explicarea procesului de efectuare a raționamentului.

• *OPS 5* este un limbaj care permite generalizarea atît a reprezentării datelor, cît și a structurii de control, compararea modelelor și interpretarea eficientă a raționamentului direct. Sistemul a fost implementat în limbajele BLISS, MACLISP și FRANZ LISP. Este unul dintre cele mai utilizate limbaje de ingineria cunoașterii, fiind cunoscut sub forma standard pentru rularea pe calculatoare DEC, precum și sub forma variantei OPS 5 e.

• *OPS 83* integrează raționamentul direct, programarea prin reguli și proceduri, fiind în esență un limbaj similar cu Pascal, la care se adaugă modul de construire a regulilor și elementelor memoriei de lucru de la OPS 5. *OPS 83* poate fi rulat pe sisteme de calculatoare electronice DEC VAX 11/750 și 11/780.



• *PERSONAL CONSULTANT* este un limbaj de reprezentare prin reguli și cadre. Sistemul are posibilitatea introducerii funcțiilor LISP definite de utilizator. *PERSONAL CONSULTANT* este scris într-un dialect al limbajului LISP, denumit IQLISP, pentru calculatoarele TI Proffesional și Explorer, precum și pentru alte microcalculatoare sub sistemul de operare MS-DOS.

• *PRISM (PR*ototype *I*nference *S*yste*M)* este un sistem care, pe baza raționamentului direct și invers, are posibilitatea să organizeze baza de date în structuri ierarhice avînd strategii de control și mecanisme de introducere a inferențelor proprii. Suportul este asigurat de programe-editor pentru crearea și întreținerea bazei de cunoștințe sub forma unor reguli, ca în limba engleză. *PRISM* este implementat în limbaj PASCAL pentru calculatoare IBM 70, sub sistemul de operare VM/CMS.

• *PSYCO (P*roduction *SY*stem *CO*mpiler*)* este un limbaj de ingineria cunoașterii pentru reprezentări prin reguli și cadre, proiectat inițial în scopul cunoașterii unor modele de procese cognitive în luarea deciziilor de către om, dar dezvoltat pentru aplicații mai largi în cercetare.

• *RADIAL* — elaborat pe baza teoriei automatelor finite — este utilizat de programele de construire a sistemului RULEMASTER. Programul constă dintr-un set de module, fiecare conținînd o rețea de stări de tranziție (fig. 1.9). Limbajul este structurat în blocuri cu variabile de scop, argumente, invocări de module și ramificări condiționale. Sistemul a fost implementat în limbajul de programare C pentru mini și microcalculatoare de tipul VAX și stații de lucru Sun, sub sistemul de operare Unix.

• *RITA (R*and *I*ntelligent *T*erminal *A*gent) contribuie la dezvoltarea cunoștințelor producătorilor de sisteme și rețele de calculatoare electronice. Sistemul este scris în limbajul de programare C pentru PDP 11/45 și 11/70, sub sistemul de operare Unix.

• *ROSIE (R*ule *O*riented *S*ystem for *I*mplementing *E*xpertise) conține o interfață la sistemul local de operare, ceea ce conduce la controlul acțiunilor la distanță. Programele suport includ operațiile de editare și depanare. El

este implementat în limbaj INTERLISP și a fost convertit în limbajul de programare C.

• *SAVOIR* cuprinde un compilator al bazei de date, un suport pentru logica Fuzzy, scheme de control prin raționament direct și invers etc. Sistemul poate fi rulat pe sisteme de tipul DEC VAX și IBM PC.

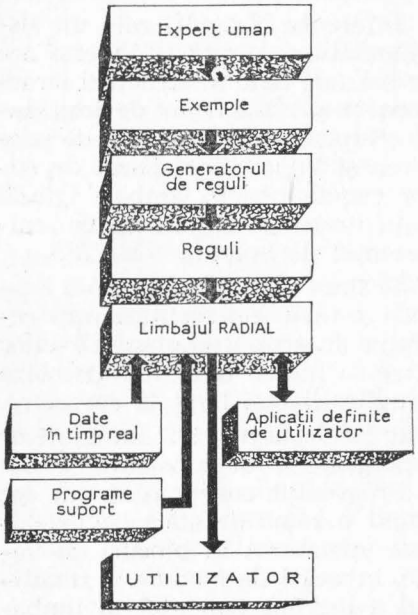



*Fig. 1.9.* Generarea regulilor din exemple și utilizarea limbajului RADIAL în sistemul RULEMASTER

• *SOAR* este o arhitectură pentru rezolvarea problemelor generale destinate reprezentării prin reguli a examinărilor de natură euristică. Sistemul asigură examinarea în spațiul problemei, care cuprinde un set de stări reprezentînd soluția și un set de operatori pentru transformarea stărilor una în alta. Crearea ierarhiilor de scopuri intermediare se face în mod automat. Inițial el a fost scris în XAPS 2, un sistem de producție paralel, apoi a fost transcris în OPS 5, cu extensie pentru rulare în paralel.

• *S.1* este un limbaj de ingineria cunoașterii pentru reprezentări prin reguli, cadre și proceduri. Principalele



caracteristici sînt : raționamentul invers, mecanismul de manipulare încorporat și blocurile de control pentru reprezentarea procedurilor. Programele-suport cuprind facilități de explicare, depanare grafică etc. *S.1* este implementat în limbaj INTERLISP pentru stații de lucru Xerox 1100 și 1108.

Descrierea programelor și limbajelor de programare pentru sistemele de inteligență artificială demonstrează marea diversitate de metode și tehnici utilizate, în funcție de aplicația avută în vedere. Cu toate că sistemele care au fost prezentate pot fi considerate de uz general, cele mai multe au apărut din necesitatea de a rezolva problemele specifice unui anumit domeniu.

# 2

# Sisteme de calcul specializate

La numai zece ani de la apariția primului calculator electronic, din știința calculatoarelor s-a desprins un nou domeniu care treptat a devenit independent — inteligența artificială (AI). Deși cu o existență puțin îndelungată — ceva mai mult de trei decenii —, inteligența artificială nu poate fi abordată, fie și numai sumar, fără a analiza în paralel și evoluția tehnicii de calcul.

Prezentînd în cele ce urmează etapele importante care au marcat dezvoltarea calculatoarelor, vom accentua asupra momentelor cruciale din evoluția acestora și, în special, asupra acelor momente în care inteligența artificială a înregistrat progrese remarcabile în principal datorită evoluției calculatoarelor, ilustrînd și aspectele referitoare la influența puternică exercitată în prezent de inteligența artificială asupra tehnicii de calcul.

După cum rezultă din numeroasele lucrări de specialitate destinate aprofundării acestui domeniu, conceptul de inteligență artificială întrunește următoarele caracteristici mai importante : posibilitatea implementării pe o mașină realizată de om ; evoluția, în sensul creșterii continue a performanțelor sistemului respectiv prin intervenția factorului uman ; autoeducarea, mașina respectivă fiind dotată cu elemente care să-i permită un dialog complex cu lumea exterioară fără intervenția omului. Într-un anumit sens, inteligența artificială s-a născut din știința



calculatoarelor și se ocupă cu proiectarea și utilizarea sistemelor inteligente de calcul.

Prin analogie cu inteligența umană, un sistem de calcul este considerat inteligent dacă înțelege vorbirea, învață, raționează, rezolvă probleme etc.

Sistemele experimentale de inteligență artificială includ programe pentru rezolvarea unor probleme din diferite științe — chimie, biologie, geologie, mecanică, medicină etc. —, elaborarea unor dispozitive de manipulare automate care să realizeze sarcini utile cu caracter repetitiv, precum și pentru translatarea unor texte în diferite limbi.

Conceptele cibernetice ale sistemelor care se autoorganizează — descrise de Ștefan Odobleja, Norbert Wiener, Mc Culloch etc. — vizează, printre altele, comportarea sistemelor la nivel macroscopic.

Începînd cu Ștefan Odobleja, ideile promovate de specialiștii ciberneticieni au influențat activitatea primilor cercetători în domeniul inteligenței artificiale. Elementul ce a asigurat unitatea tuturor acestor idei a fost constituit de dezvoltarea mașinii de calcul, începută de Charles Babbage, continuată de Alan Turing, Von Neumann și alții.

Majoritatea cercetărilor care vizează progresul inteligenței artificiale sînt concentrate pe următoarele aspecte : rezolvarea problemelor, raționamentul logic, limbaj, programe — programare automată, instruire, expertiză, robotică și sisteme de vedere.

Un calculator este un sistem complex ce încorporează un număr variabil de subansamble, realizate în diferite tehnologii, pe care se pot executa diferite operații ce permit creșterea posibilităților oferite de creierul uman.

Încercarea de a descrie sistemele de inteligență artificială presupune prezentarea acestora pornind de la simplu la complex, în sensul că dacă un sistem natural are ca element fundamental celula, continuă apoi cu părțile componente — organele — și ajunge în final la organism, tot așa și cel artificial, pornind de la circuitele simple, ajunge la sistemul complex de inteligență arti-

ficială. Pentru un sistem, considerat un tot unitar, sau pentru fiecare componentă este necesar să distingem următoarele caracteristici : funcția, performanța și structura, elemente care reprezintă criterii după care se poate realiza clasificarea sistemelor de calcul.

## 2.1. STRUCTURA SISTEMELOR DE CALCUL ACTUALE

Apariția primelor calculatoare electronice către jumătatea secolului XX — ENIAC și EDVAC (S.U.A.), EDSAC (Anglia) — a fost pregătită de numeroase invenții elaborate pe parcursul a mai bine de trei veacuri. Printre acestea, în succesiunea timpului, se numără mașinile mecanice de calcul ale lui Blaise Pascal și Gottfried W. Leibnitz, mașina de calcul analitic a lui Charles Babbage, dispozitivele de citit cartele realizate de Hollerith și, mai tîrziu, calculatoarele cu relee construite în Laboratoarele Bell și la Universitatea Harvard.

În ultimii patruzeci de ani evoluția calculatoarelor electronice a fost atît de rapidă și spectaculoasă, încît conceptul de generație a fost adoptat în mod ferm pentru acest domeniu. În tabelul 2.1 sînt prezentate principalele caracteristici hardware și software ale generațiilor de calculatoare cunoscute pînă în prezent.

În lucrarea *Prima schiță de raport asupra EDVAC*, John von Neumann stabilește structura de bază a calculatorului cu program memorat. Schema-bloc a calculatorului conceput de von Neumann este ilustrată în fig. 2.1.

După 1950 au fost produse sute de mii de calculatoare, toate fiind realizate după principiile lui von Neumann. În perioada anilor '50—'60, datorită descoperirilor în electronică, memoria calculatorului a devenit mai ieftină și mai rapidă : în paralel, a fost dezvoltat și conceptul de stivă, iar unitățile aritmetice și logice au devenit tot mai sofisticate.



Dispozitivele de intrare/ieșire — de la mașina de scris pînă la discuri magnetice, benzi magnetice, terminale la distanță etc. — au cunoscut o continuă perfecționare. Cu toate acestea părțile componente ale unui sistem de calcul convențional — unitatea de control *(UC)*, unitatea aritmetică și logică *(UAL)*, memoria *(M)* și dispozitivele de intrare/ieșire *(I/E)* — au fost integral prezentate, într-o formă sau alta, în structura calculatoarelor. Organizarea sistemelor de calcul după principiile lui Von Neumann s-a impus după anul 1960, perioadă în care au apărut noi componente electronice care au permis, în special, creșterea vitezei de operare a calculatoarelor electronice.

Viteza de operare a circuitelor electronice este în ultima instanță limitată de viteza luminii (lumina parcurge aproximativ 30 cm într-o nanosecundă) ; ca urmare circuitele integrate operează în domeniul nanosecundelor. Creșterea vitezei de operare a circuitelor electronice se realizează din ce în ce mai dificil și cu cheltuieli tot mai mari.

Cerințele mereu sporite ale utilizărilor au determinat apariția calculatoarelor specializate pentru anumite activități ; de exemplu, pentru calcule științifice se folosesc calculatoare de capacitate mare, iar pentru educație și învățămînt pot fi utilizate microcalculatoare.

În tabelul 2.2 sînt prezentate o parte din funcțiile calculatoarelor și tipurile de calculatoare care pot satisface aceste funcții.

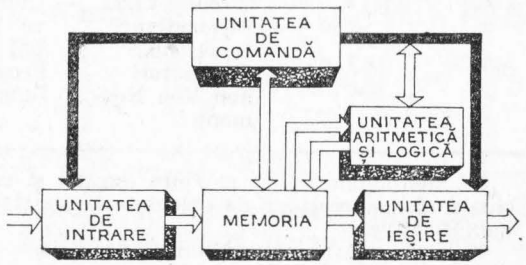



*Fig. 2.1.* Schema-bloc a calculatorului numeric.

TABELUL 2.1. **Generații de calculatoare electronice**

| Generația | Perioada | Hardware | Software | Viteza de operare (Kips)** |
|---|---|---|---|---|
| I | 1946—1957 | — Tuburi electronice<br>— Memorii pe tambur magnetic | — Prelucrare serie<br>— Concept de programare | 10 |
| a II-a | 1957—1964 | — Diode<br>— Tranzistoare<br>— Memorii din ferite | — Monitorizare<br>— Compilare<br>— Multiprogramare | 200 |
| a III-a | 1964—1971 | — Memorii pe disc magnetic<br>— Circuite integrate<br>— Utilizarea liniilor de comunicație<br>— Memorie semiconductoare | — Sisteme de operare | 5 000 |
| III—IV | 1971—1974 | — Circuite SSI, MSI<br>— Memorii RAM, PROM, REPROM | — Soft, implementat în hard<br>— Limbaje de nivel înalt | 15 000 |
| a IV-a | 1974—1981 | — Circuite LSI, VLSI<br>— Circuite ECL<br>— Memorii pe bule magnetice<br>— Discuri optice | — Limbaje de nivel foarte înalt<br>— Limbaje orientate pe obiecte | 30 000 |
| a V-a | 1981 * | — Circuite VLSI<br>— Dispozitive funcționale<br>— Arhitecturi non Von Neumann | — Comunicare în limbaj natural<br>— Programare funcțională | 1 000 000 |

* Menționăm că în privința duratei și caracteristicilor principale ale generațiilor de calculatoare există unele deosebiri nesemnificative.

** Kips — kilo-instrucțiuni pe secundă.



TABELUL 2.2 **Tipuri de calculatoare și funcțiile acestora**

| Funcția calculatorului | Tipul calculatorului |
|---|---|
| Științifică | Calculatoarele de capacitate mare și supercalculatoare |
| Comercială<br>Gestiune economică<br>Comunicații<br>Educație și învățămînt | Minicalculatoare |
| Divertisment | Microcalculatoare și calculatoare pe o structură |
| Inteligență artificială | Calculatoare din generația a cincea |

Datorită evoluției tehnologiei, calculatoarele, indiferent de tip, au avut o structură asemănătoare cu toate diferențele datorate destinației pe care o aveau.

Prezentăm în tabelul 2.3 evoluția calculatoarelor din punct de vedere tehnologic, subliniind și generația care a utilizat respectiva tehnologie.

TABELUL 2.3 **Evoluția calculatoarelor din punct de vedere tehnologic**

| Tehnologia | Generația | Complexitatea componentelor | Viteza de operare (s) |
|---|---|---|---|
| Mecanică | — | — | — |
| Electromecanică | — | — | $10^{-1}$ |
| Tuburi cu vid | I | — | $10^{-3}$ |
| Tranzistoare | a II-a | un tranzistor | $10^{-5}$ |
| Circuite hibride | | | |
| Circuite integrate SSI, MSI | a-III-a | 1—10 porți SSI<br>10—100 porți MSI | $10^{-7}$ |
| Circuite integrate LSI și VLSI | | 10—10 000 porți LSI<br>10 000—100 000 porți | $10^{-8}$ |
| Circuite integrate ULSI | a IV-a | VLSI | |
| | a-V-a | 100 000 porți | |

În fig. 2.2 sînt redate tipurile reprezentative de calculatoare, de la primul calculator electronic și pînă în prezent.

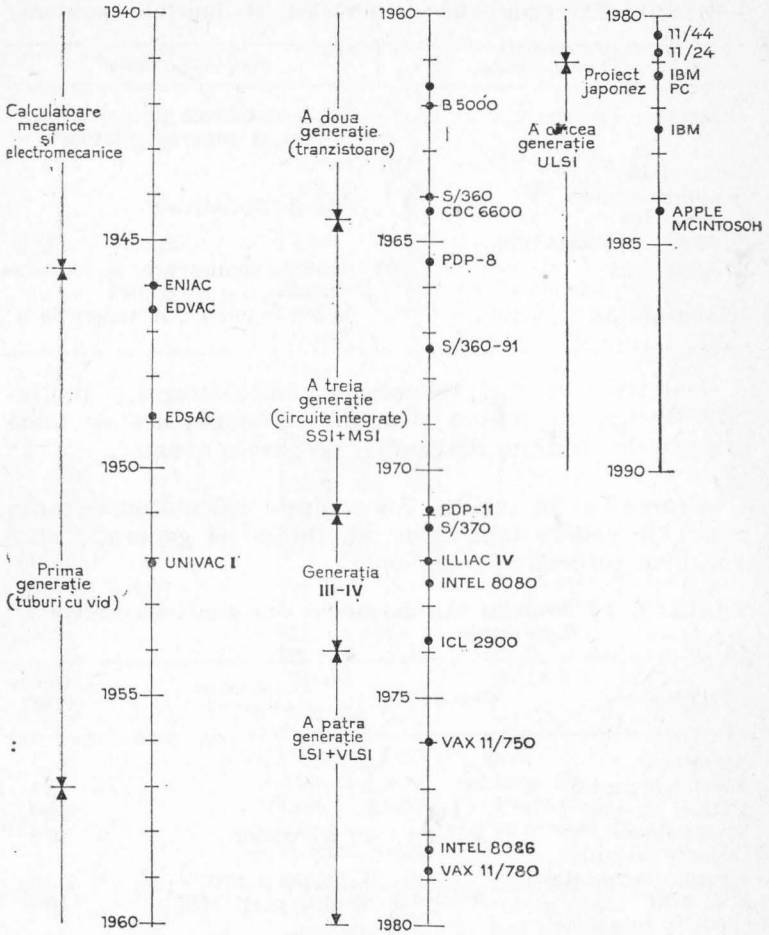



*Fig. 2.2.* Tipuri reprezentative de calculatoare din cele cinci generații.

## 2.1.1. SISTEMUL DE CALCUL B 5000

În general, la proiectarea unui sistem de calcul, operația prioritară constă în realizarea structurii hardware, partea de software fiind astfel elaborată încît să permită



în final utilizarea cu succes a hardware-ului respectiv. În contrast cu acest mod de lucru, calculatorul B 5000 a fost conceput ca un ansamblu hardware-software, fiind destinat lucrului cu limbaje de nivel înalt și controlului evoluției propriilor sale activități.

Proiectanții acestui sistem au urmărit să mărească viteza de operare și să realizeze o mai mare flexibilitate hardware și software. Pentru atingerea acestor obiective s-a ținut seama de următoarele aspecte : exprimarea problemelor în limbaj de nivel înalt ; viteza de operare a limbajului-mașină ; programul de depanare a limbajului de nivel înalt ; timpul de pornire și încărcare ; eficiența sistemului de operare ; facilități în întreținerea și în realizarea de modificări în programele existente ; ușurința reprogramării în cazul schimbării configurației sistemului.

Criteriile care s-au aflat la baza proiectării sistemului de calcul B 5000 au fost următoarele : programul să fie independent în zona de memorie alocată și nemodificabil în timpul rulării; independența datelor în memorie ; adresarea memoriei în cadrul unui program să se bazeze pe schema de adresare contextuală care permite reducerea redundanței ; sintaxa programului să permită o translatare mecanică din limbajul-sursă într-un cod-mașină eficient ; realizarea de facilități care să permită controlul de ansamblu al sistemului asupra propriilor sale operații; separarea activității de intare/ieșire de activitatea procesorului central ; existența unui sistem de operare eficient ; multiprogramarea și procesarea paralelă.

Organizarea calculatorului B 5000 este modulară, subansamblele fiind conectate prin intermediul unor switch-uri electronice asemănătoare celor din telefonie. În fig. 2.3 se prezintă schema-bloc pentru o configurație completă a calculatorului B 5000.

Sistemul este însoțit de un program principal de control înregistrat în memoria externă. În timpul operațiilor normale, o mică porțiune din acest program este transferată în memoria internă. Această secvență permite manipularea unui procent important de operații recurente. Alte segmente ale acestui program sînt aduse în memoria internă de pe discul magnetic ori de cîte ori este necesar, pentru manipularea evenimentelor care se petrec mai rar

cînd sistemul execută programul principal de control sau se află în starea de control, CS *(control state)*.

Orice intrare în starea de control se execută prin intermediul întreruperilor. Dacă sistemul se află în starea de control se execută o operație specială care permite

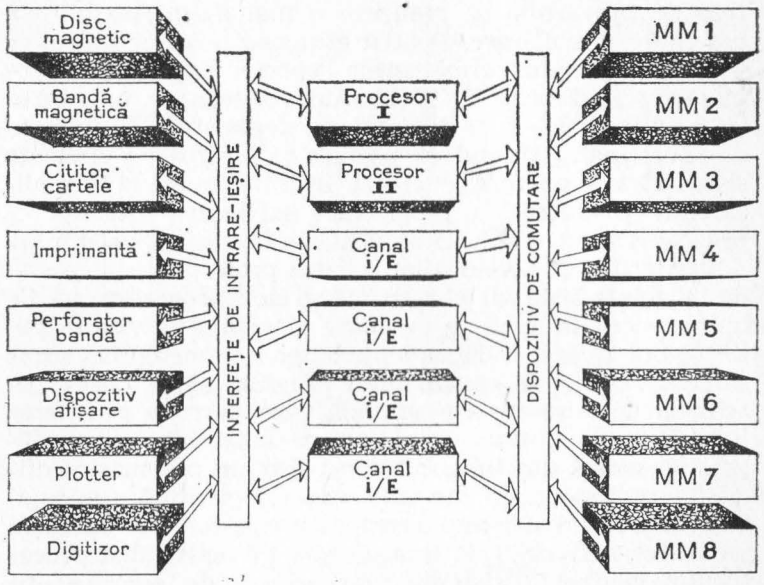



*Fig. 2.3.* O configurație completă a sistemului de calcul B 5000.

controlului să se întoarcă la programul-obiect ce se execută în momentul apariției întreruperii, care poate fi cauzată de următoarele evenimente : disponibilitatea unui canal de intrare/ieșire, terminarea unei operații de intrare/ieșire sau apariția unei operații de indexare care a violat protecția memoriei.

Pentru procesarea condițiilor de întrerupere, programul principal de control manipulează părțile fundamentale ale sistemului de operare.

Sistemul B 5000 poate opera cu cuvinte de lungime fixă sau de lungime variabilă. Aceste moduri de lucru poartă denumirea de mod-cuvînt și, respectiv, mod-caracter, ele fiind folosite pentru realizarea unor operații



specifice. Prin combinarea ambelor posibilități într-un singur procesor, se poate alege modul de lucru în funcție de necesități.

În sistemul B 5000 este posibil ca un procesor să lucreze în mod-cuvînt, iar al doilea în mod-caracter.

Pentru acest calculator programele sînt alcătuite din șiruri de silabe. Silaba, avînd o lungime fixă de 12 biți, este unitatea de bază a programului. Termenul de silabă este folosit în locul celui de instrucțiune, pentru a putea face deosebirea cuvenită între silabe și instrucțiunile care au lungimi diferite.

Fiecare cuvînt-program conține silabe ; acestea sînt excutate în ordinea de la stînga la dreapta în cadrul cuvîntului-program, iar cuvintele sînt executate secvențial în cadrul programului. Saltul este permis la orice silabă din cuvînt.

Organizarea internă a calculatoarelor care adresează o singură locație de memorie la un moment dat determină pierderea unor perioade importante de timp cu memorarea și readucerea rezultatelor intermediare în secvența de calcul. Datele trebuie plasate în registre și în locațiile de memorie înainte de executarea operației. În ceea ce privește conținutul lor, acestea trebuie adesea reorganizate înaintea lansării operației următoare.

Calculatoarele multiadresă sînt destinate executării cît mai eficiente a anumitor operații. Dintre dezavantaje menționăm prețul destul de ridicat al echipamentului. Pe scurt, proiectarea calculatoarelor convenționale, cu sau fără ajutorul programării automate, necesită un mare efort de programare, o capacitate mare a memoriei și mult timp pentru depășirea limitelor propriei organizări interne. La calculatorul B 5000 se încearcă rezolvarea acestor probleme prin încorporarea unei stive push-down, care elimină complet instrucțiunile de memorare și de readucere a rezultatelor intermediare.

În cazul sistemului analizat stiva este compusă din următoarele elemente : o pereche de registre, *A* și *B*, și o zonă de memorie. Cînd operandul este adus de către program el este plasat în registrul *A* ; dacă registrul *A* conține un cuvînt de informație, atunci acesta este trans-

ferat în registrul *B* înainte de încărcarea operandului în registrul *A*. Dacă registrul *B* este de asemenea ocupat cu o informație, atunci cuvîntul din registrul *B* este transferat în zona de memorie definită de către un registru de adresă *S*. În acest caz cuvîntul din *A* este transferat în *B*, iar în registrul *A* poate fi adus operandul. Noul cuvînt sosit în stivă determină creșterea cu o unitate a adresei din registrul *S*.

Informația conținută în registre este ultima informație intrată în stivă. Stiva operează pe principiul LIFO (*Last In First Out*). Logicianul polonez J. Lukasiewicz a dezvoltat un mod de notare pentru scrierea expresiilor algebrice și logice care nu necesită alternarea operatorilor cu simbolurile ce intră în cadrul unor operații, așa cum se procedează în cadrul convențiilor precedente. Acest mod de notare — bazat pe plasarea operatorilor în dreapta operanzilor — este folosit pentru calculatorul B 5000 ; așadar $A+B$ devine $AB+$, iar $A+B+C$ poate fi scris $AB+C+$ sau $ABC++$.

Unul din scopurile urmărite de producătorii calculatorului B 5000 se referă la realizarea independenței adresării. Prin segmentarea automată a programelor a rezultat posibilitatea de a avea programe a căror dimensiune este independentă de mărimea memoriei interne.

Fiecare segment al programului și fiecare zonă de date ar trebui să fie complet relocabile fără modificări ale programului. Este astfel posibil să se încarce toate segmentele programului sau programul integral pe memoria magnetică, ele putînd fi readuse în memoria internă, în zonele disponibile, atunci cînd este cazul. Datorită densității foarte mari a programelor din sistemul B 5000 (aproximativ 60 000 de instrucțiuni cu un singur operand) memoria externă trebuie să asigure capacitatea cerută. În consecință, sistemul pe care se rulează astfel de programe va fi de complexitate apreciabilă ; viteza de procesare devine funcție de dimensiunea memoriei interne. Dacă programele de dimensiuni considerabile sînt rulate pe sisteme de calcul cu memorie internă redusă, timpul de lucru va fi consumat pentru readucerea segmentelor de program din memoria externă în cea internă. În cazul cînd memoria internă este extinsă, atunci timpul afectat



unor astfel de activități va fi mai redus și, în consecință, viteza de rulare va crește corespunzător.

În concluzie, din analiza calculatorului B 5000 se desprind următoarele caracteristici importante : prezența stivelor ; specificarea tipului de date prin adăugarea unui indicator datei ; maparea multiprogramului (pentru accesarea scalarilor, vectorilor sau matricelor se folosesc descriptori ; această tehnică de adresare indirectă permite multiprogramarea) ; multiprocesarea (un sistem B 5000 poate fi construit cu două procesoare centrale) ; utilizarea a două moduri de lucru — modul-caracter și modul-cuvînt.

### 2.1.2. SISTEMUL ILLIAC IV

În faza actuală a dezvoltării sistemelor de calcul au apărut două noi modalități de creștere a performanțelor calculatoarelor electronice — suprapunerea și duplicarea.

În primul caz structura hardware a fost astfel modificată încît operațiile executate de două sau mai multe componente funcționale pot să se suprapună, ceea ce are drept consecință reducerea timpului total de operare. Precizăm că în perioada premergătoare executării operațiilor care pot fi suprapuse secvențele de control între componente trebuie să fie decuplate. În mod sigur unitatea de control poate să aducă instrucțiunea următoare, în timp ce unitatea aritmetică și logică execută pe cea prezentă.

În al doilea caz, una din cele patru mari structuri componente ale sistemului de calcul — UC, UAL, M, I/E — poate exista în mai multe exemplare. Menționăm faptul că, în cadrul calculatoarelor electronice, duplicarea dispozitivelor de I/E s-a realizat de timpuriu, echipamentele respective fiind dotate cu mai multe benzi magnetice, discuri, cititoare de cartele etc.

Suprapunerea și duplicarea nu se exclud reciproc, o a treia posibilitate rezultată din combinația celor două, impunîndu-se treptat. Precizăm, de asemenea, că tehnica suprapunerii a fost implementată prin memorii-tampon și mecanismul de tip *pipeline* (fig. 2.4).

Tehnica pipeline asigură divizarea unei operații în suboperații sau faze, separîndu-le una de alta, astfel încît fazele odată separate pot fi executate simultan. Mecanismul memoriei-tampon permite ca o operație să fie divizată în operații paralele prin crearea unui spațiu de memorie pentru controlul informației.

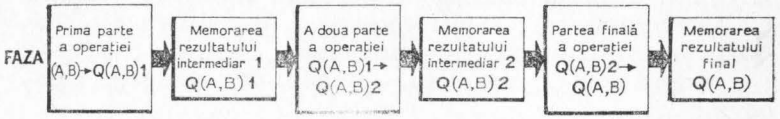

FAZA
Prima parte a operației (A,B)→Q(A,B)1
Memorarea rezultatului intermediar 1 Q(A,B)1
A doua parte a operației Q(A,B)1→ Q(A,B)2
Memorarea rezultatului intermediar 2 Q(A,B)2
Partea finală a operației Q(A,B)2→ Q(A,B)
Memorarea rezultatului final Q(A,B)

*Fig. 2.4.* Prezentarea mecanismului de pipeline.

Tehnica duplicării este folosită de multiprocesorul general, dispozitiv complex ce permite multiplicarea a trei din cele patru mari părți componente ale sistemului.

Costul multiprocesorului este foarte ridicat și se consideră că eventualele opțiuni ar afecta performanțele sistemului. Din aceste motive au fost preferate anumite opțiuni particulare, constînd numai în modificarea uneia din cele trei structuri componente care au fost multiplicate anterior : memoria, unitatea aritmetică și logică sau unitatea de control.

Centralizarea unității de control dă naștere la organizarea de bază a procesorului vectorial, așa cum se întîmplă în sistemul ILLIAC IV.

Opțiunile particulare au fost alese din două motive principale :

*costul* — Este cunoscut faptul că o parte importantă din prețul unui calculator este atribuit unității de control. Multiplicarea acestei componente este destul de costisitoare, motiv pentru care se preferă centralizarea unităților de control, modalitate mai rentabilă decît centralizarea celorlalte două componente (memoria și unitatea aritmetică și logică) ;

*structura* — Există o gamă largă de probleme științifice și economice care pot fi rezolvate cu un calculator electronic prevăzut cu o singură unitate centrală (cu un șir de instrucțiuni) și mai multe unități aritmetice și logice.



Același algoritm este repetat pe mai multe seturi de date diferite, acestea fiind organizate sub forma unui vector ; procesorul vectorial al calculatorului ILLIAC IV operează cu vectorul de date. Toate componentele de date structurate ca un vector sînt procesate în paralel.

Proiectul pentru realizarea calculatorului ILLIAC IV a fost elaborat în cadrul Universității Illinois. Se urmărea, ca obiectiv principal, dezvoltarea unui sistem de calcul care să utilizeze principiul operării în paralel, pentru a atinge viteza de operare de peste 1 miliard de instrucțiuni pe secundă. Pentru atingerea acestei performanțe, sistemul trebuia să folosească un număr de 256 de procesoare divizate în patru subansamble a cîte 64 de procesoare fiecare — care să opereze simultan sub un control central.

Datorită dificultăților de ordin tehnologic și a complexității circuitelor logice necesare, cheltuielile au crescut enorm, astfel încît a fost finalizat numai un sfert din proiect.

Acest dispozitiv, cu 64 de procesoare, atinge o viteză de peste 200 milioane instrucțiuni pe secundă.

Performanțele sistemului ILLIAC IV au depășit limitele fundamentale în ceea ce privește viteza de calcul, demonstrînd practic posibilitatea ca un număr nelimitat de calcule să se realizeze simultan.

Proiectul calculatorului ILLIAC IV, la nivel de schemă logică, este inspirat după un calculator construit înainte de 1960 de un grup de cercetători de la Westinghouse Electric Corporation. În acest proiect o singură unitate de control (CU — *C*ontrol *U*nit) trimite instrucțiuni către un număr mare de elemente de procesare (PE — *P*rocessing *E*lement) și adrese către unitățile individuale de memorie (PEM — *P*rocessing *E*lement of *M*emory) asociate elementelor de procesare. Așadar, controlul este asigurat de către o singură secvență de program. Aceasta controlează un număr de elemente de procesare care execută simultan aceeași instrucțiune asupra datelor ce pot fi diferite în memoriile asociate în parte elementelor de procesare.

Fiecare din cele 64 de elemente de procesare ale calculatorului ILLIAC IV este de fapt o puternică unitate

de calcul, ce poate realiza o gamă largă de operații aritmetice asupra numerelor cu lungime de 64 biți. Aceste numere pot fi reprezentate în unul din următoarele formate : unitar (cu un singur număr cu lungimea de 64 de biți) sau „spart" (mai multe numere mai mici, dar de dimensiuni egale).

Unitatea de memorie are o capacitate de 2 048 de locații de 64 de biți. Timpul de acces la aceste memorii este de 188 ns, dar din cauza dificultăților ce pot apare în cazul existenței mai multor cereri de acces simultan timpul minim între operații succesive de memorie este mărit pînă la 350 ns.

Fiecare element de procesare conține peste 100 000 de componente electronice distincte, asamblate în aproape 12 000 circuite de comutare. Un element de procesare împreună cu unitatea sa de memorie și logică asociată poartă numele de unitate de procesare (PU — *P*rocessing *U*nit).

Într-un sistem care conține mai mult de 6 milioane de componente ne putem aștepta ca o componentă sau o conexiune să se defecteze în cîteva ore. Din acest motiv o atenție deosebită s-a acordat procedurilor de diagnosticare și testare.

Fiecare din cele 64 unități de procesare sînt permanent supuse în mod automat unui set de teste. Dacă o unitate este găsită necorespunzătoare de către acest test subansamblul defect se înlocuiește imediat cu o pierdere minimă din timpul de operare. Cînd o unitate de procesare este defectă ea este remediată separat prin intermediul unui calculator de diagnosticare și depanare. Odată pana descoperită și remediată, unitatea de calcul continuă operațiile cu posibilitățile maxime.

ILLIAC IV nu putea fi proiectat și realizat fără ajutorul altor calculatoare. Calculatorul de capacitate medie Burroughs 6 500 a lucrat permanent timp de doi ani pentru pregătirea listelor de cablaj ale circuitelor imprimate, ale sertarelor precum și pentru dezvoltarea programelor de test și diagnoză.

Proiectul original al lui ILLIAC IV, ca sistem general de procesare a vectorilor, constă în faptul că unitatea de comandă este decuplată de restul procesorului, astfel



încît anumite instrucțiuni pot fi executate complet numai cu resursele unităților de control în timp ce unitatea aritmetică și logică realizează operațiile vectoriale. În acest fel se remarcă un alt grad de paralelism, ce se însumează la paralelismul inițial al celor 64 de unități aritmetice și logice conduse simultan. În sistemul ILLIAC IV există două calculatoare — unul care operează cu mărimi scalare, iar altul care operează cu vectori. Toate instrucțiunile pornesc din calculatorul care operează cu mărimi scalare.

Fiecare element al matricei de unități aritmetice și logice constituie un element de procesare ; există 64 de astfel de elemente, numerotate de la 0 la 63, fiecare răspunzînd la instrucțiunile pe care le primește dacă este un modul activ. Trebuie menționat faptul că există instrucțiuni care activează sau dezactivează elementul de procesare. Acesta realizează aceeași operație sub comanda unității de control. Elementele de procesare sînt însoțite de unități de memorie PEM cu o capacitate de 2 048 cuvinte de 64 biți cu timpul de acces de 350 ns.

Pentru deplasarea informațiilor între PEM se folosesc seturi speciale de instrucțiuni. În sistemul de calcul ILLIAC IV datele pot fi reprezentate în formate diferite. Prin utilizarea formatelor de date cu lungime de 64, 32 și 8 biți, cele 64 elemente de procesare pot prelucra vectori de operanzi cu 64, 128 sau 512 componente.

Astfel ILLIAC IV poate aduna 512 operanzi de 8 biți în aproximativ 66 ns, fiind capabil să realizeze aproape $10^{10}$ adunări într-o secundă. De asemenea sistemul ILLIAC IV poate realiza într-o secundă 150 milioane de adunări pentru cuvinte de 64 biți.

Dispozitivele de intrare/ieșire sînt controlate de un sistem de calcul B 6500. Sistemul de operare include asambloare și compilatoare rezidente în calculatorul B 6500.

În fig. 2.5 este prezentată organizarea sistemului ILLIAC IV. Sistemul este compus din matricea sistemului și subsistemului de intrare/ieșire. Matricea sistemului este alcătuită din procesorul matriceal și unitatea de comandă. Procesorul matriceal are 64 PE și 64 PEM asociate.

Sistemul de intrare/ieșire al sistemului ILLIAC IV este alcătuit din subsistemul de I/E, subsistemul de disc magnetic și calculatorul B 6500. Sistemul de calcul ILLIAC IV este unul din cele mai performante realizate pînă în prezent.

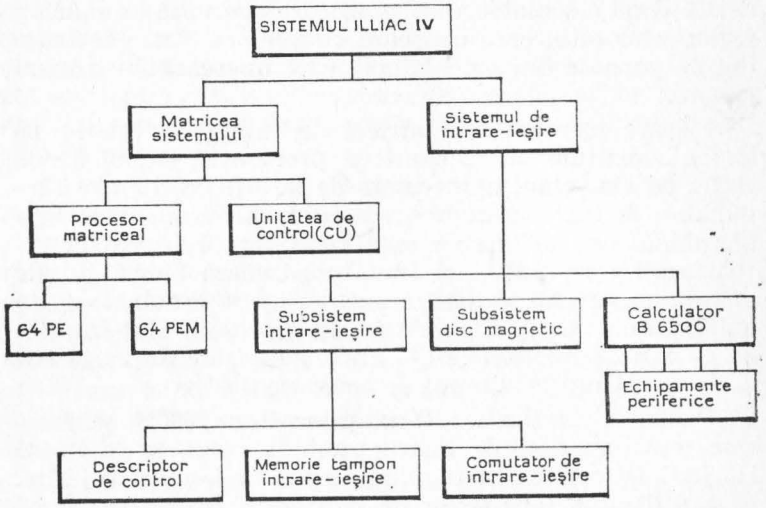



*Fig. 2.5.* Organizarea sistemului ILLIAC IV.

## 2.1.3. SISTEMUL CONTROL DATA 6600

În vara anului 1960, firma Control Data a inițiat un proiect care a fost finalizat în octombrie 1964, odată cu livrarea primului calculator de tip 6600. În acea perioadă se aprecia că performanțele circuitelor electronice și operarea paralelă ar fi două caracteristici importante pentru orice calculator avansat.

Sistemul de calcul 6600 a fost gîndit pentru a fi utilizat în rezolvarea unor probleme științifice de mare amploare și pentru soluționarea în time-sharing (timp divizat) a unui număr ridicat de probleme de dimensiuni mai reduse. În primul caz a fost necesară elaborarea unui



procesor în virgulă mobilă de mare viteză cu acces direct la memoria calculatorului.

La construirea calculatorului 6600 s-a impus cu necesitate separarea activității procesorului central de activitatea echipamentelor periferice. De aici s-a născut ideea realizării mai multor procesoare periferice, după cum se poate vedea în fig. 2.6. Cele zece procesoare periferice ale sistemului 6600 au acces la memoria centrală și la canalele periferice.

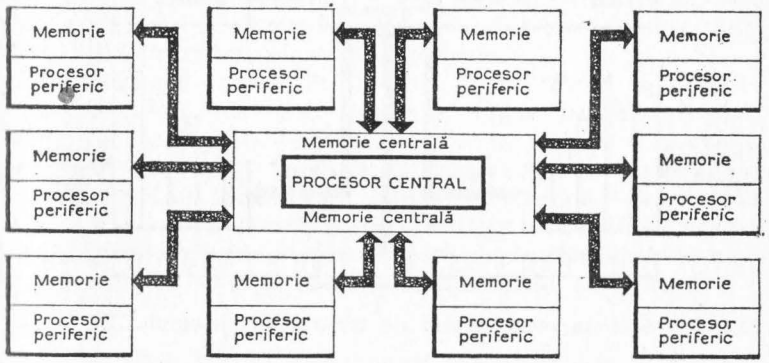



*Fig. 2.6.* Schema-bloc a sistemului CDC 6600.

Controlul executiv al sistemului este întotdeauna asigurat de unul din procesoarele periferice, celelalte operînd cu periferiicele asignate cu task-urile de control.

Toate cele zece procesoare au acces la douăsprezece canale de intrare-ieșire, precum și la memoria centrală ; ele pot efectua transferuri de informații cu această memorie. Fiecare din cele zece procesoare conține propria sa memorie pentru programe și zone tampon.

Fiecare procesor are o memorie de 4 096 cuvinte de 12 biți. Formatul instrucțiunilor este de 12 și 24 biți pentru a permite adresarea directă, indirectă și relativă. Instrucțiunile realizează operații logice, adunarea, scăderea și salturile condiționate. Instrucțiunile asigură, de asemenea, transferurile de informații cu oricare din cele 12 canale periferice și pot transfera un cuvînt sau un bloc în/din memoria centrală.

Cuvintele din memoria centrală, cu o lungime de 60 de biți, sînt asamblate din cinci cuvinte, fiecare a cîte 12 biți. Procesoarele periferice posedă instrucțiuni capabile să interpună activitatea procesorului central.

Pentru obținerea unei puteri mari de calcul cu economii de materiale și spațiu a fost adoptat un proiect care folosește time-sharing-ul (fig. 2.7).

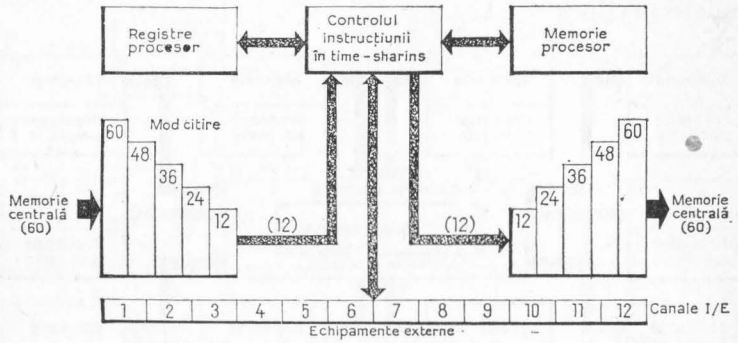



*Fig. 2.7.* Schema procesorului de control în sistemul CDC 6600.

Sistemul conține un registru asemănător unui cilindru pe care se deplasează dinamic informația pentru toate cele zece procesoare. Astfel de informații — adresa programului, conținutul acumulatorului etc. —, totalizînd 52 de biți, sînt deplasate în lungul acestui cilindru. Fiecare deplasare completă pe cilindru necesită un ciclu major sau 1000 ns. O secțiune în cilindru conține sumatoare și alte elemente care asigură realizarea unui pas pentru oricare instrucțiune de la procesoarele periferice.

Timpul necesar pentru realizarea unui pas reprezintă un ciclu minor și este egal cu 100 ns.

Fiecare din cele zece procesoare, deci, dispune de un ciclu minor din cele zece ale unui ciclu major pentru a realiza un pas. O instrucțiune poate fi executată în unul sau mai mulți pași în funcție de specificul ei. Canalele de intrare/ieșire sînt bidirecționale, avînd o lățime de 12 biți. Un cuvînt de 12 biți poate fi deplasat la fiecare ciclu major (1 000 ns) pe oricare canal.



Așadar, atingerea ratei maxime — de 120 milioane biți pe secundă — este posibilă utilizînd toate cele zece procesoare periferice. O rată de transfer de aproximativ 50 de milioane biți pe secundă poate fi menținută practic în sistemele funcționale. Fiecare canal deservește cîteva dispozitive periferice și realizează interfața cu alte sisteme, cum ar fi calculatoarele satelit. Procesoarele periferice și de control au acces la memoria centrală prin rețele specializate.

Memoria centrală este organizată în bancuri independente, cu posibilitatea de a transfera cuvinte la fiecare ciclu minor. Procesoarele periferice afectează foarte puțin controlul adresării memoriei centrale.

Procesorul central 6600 poate fi considerat drept unitate aritmetică de mare viteză a sistemului. Programul, operanzii și rezultatele sînt stocate în memoria centrală. Între procesorul central și procesoarele periferice nu există legături decît prin memorie și două semnale de control — unul pentru întreruperile procesorului central de la periferice și altul care permite controlul adreselor programului central de către un procesor periferic.

### 2.1.4. SISTEMUL DE CALCUL CRAY-1

Între procesoarele vectoriale realizate pînă în prezent se remarcă sistemul de calcul CRAY-1 realizat de Saymour-Cray. Pe lîngă acest procesor sînt cunoscute peste zece tipuri de astfel de dispozitive. Dintre ele se remarcă ILLIAC IV — cel mai puternic —, ASC (*Advanced Search Computer*) — realizat de firma Texas Instruments — și STAR-100 realizat de Control Data Corporation.

CRAY-1, cu o putere de calcul echivalentă cu puterea de calcul a cinci calculatoare de tip IBM 370/195s, este unul din cele mai puternice procesoare vectoriale elaborate pînă în prezent.

Studii de performanță realizate în ultimii ani demonstrează că în acest sistem se pot atinge în virgulă mobilă viteze de aproape 250 milioane de operații pe secundă. Performanțele sistemului se datorează în principal modului de realizare a arhitecturii interne, adaptată

în special nevoilor de calcul (calculele se fac secvențial, la fiecare pas obținîndu-se rezultate folosite la pașii următori). Prin intermediul unei tehnici denumite „de înlănțuire" unitățile vectoriale din sistemul CRAY-1 funcționează în combinație cu registrele scalare și vectoriale. În acest mod sînt generate rezultate intermediare ce se folosesc imediat fără referiri la memoria adițională, care altfel ar determina scăderea vitezei de calcul.

Desigur, există și alte elemente care au permis sporirea vitezei de calcul a sistemului CRAY-1. Printre acestea menționăm dimensiunile reduse ale cablajelor, fapt care diminuează distanțele parcurse de semnalele electronice în interiorul calculatorului, permițîndu-se utilizarea unui ceas intern cu o perioadă de 12,5 ns. Trebuie subliniat faptul că sistemul CRAY-1 este dotat cu cel mai rapid procesor scalar — o memorie semiconductoare de 1 milion de cuvinte, echipată cu logică de detectare și corectare a erorilor —, precum și cu un compilator Fortran optimizat. Precizăm, totodată, că lungimea cuvîntului este de 64 biți. Pentru a reduce la minimum lungimea cablajelor, calculatorul CRAY-1 a fost realizat sub formă cilindrică. Sistemul este echipat cu 12 canale de intrare/ieșire, 16 bank-uri de memorie, 12 unități funcționale și registre cu o capacitate de mai mult de 4 kocteți.

Accesul la memorie este asigurat de către canalele de intrare/ieșire și registrele de mare viteză. Cele mai importante caracteristici ale sistemului CRAY-1 sînt : utilizarea a numai patru tipuri de circuite integrate — circuite bipolare pentru registre de $16 \times 4$ biți cu timpul de acces de 6 ns, circuite bipolare pentru memorie $1\,024 \times 1$ bit cu timpul de acces de 50 ns, circuite logice bipolare cu timp de propagare mai mic de o nanosecundă și circuite ECL —, viteza memoriei principale, sistemul de răcire și unitatea de calcul.

Circuitele logice sînt porți de mică sau mare viteză și circuite de tip ECL. Sistemul este extrem de compact, densitatea circuitelor pe o placă de circuit imprimat fiind foarte mare ; de asemenea numărul plăcilor este destul de ridicat. Numărul extrem de mare de componente în volumul redus al sistemului ridică probleme deosebite pentru răcire.



Memoria sistemului CRAY-1 este organizată în 16 bank-uri cu 72 de module pe fiecare bank. Fiecare modul contribuie cu 1 bit la cuvîntul de 64 biți. Ceilalți 8 biți sînt folosiți pentru detectarea erorilor de paritate.

Cele 12 unități funcționale ale sistemului sînt organizate în 4 grupe : adrese, scalari, vectori și virgulă mobilă.

Tehnica „înlănțuirii" avantajează efectuarea operațiilor paralele de către unitățile funcționale. Operațiile paralele pe vectori permit obținerea a două sau mai multe rezultate pe un tact.

O operație vectorială poate utiliza, ca surse pentru operanzi, două registre vectoriale sau un registru scalar și unul vectorial. Vectorii care depășesc 64 elemente sînt procesați în segmente de cîte 64 de elemente.

Proiectul calculatorului CRAY-1 a fost influențat de experiența cîștigată ca urmare a realizării procesoarelor vectoriale din prima generație. CRAY-1 este un sistem pentru procesarea vectorilor scurți. În fine, ca un ultim considerent sistemul poate efectua operații cu mărimi scalare și vectoriale.

### 2.1.5. CALCULATORUL IBM SYSTEM/360

Structura logică, procesarea eficientă și versatilitatea sînt deservite, în calculatorul SYSTEM/360, de acumulatoare multiple cu adresare binară, operații pe bit, indexare automată, cîmpuri de lungime fixă și variabilă etc. Controlul programului de întreruperi, protecția memoriei și stările unității centrale contribuie în mod hotărîtor la o operare efectivă.

Adresarea registrelor, interfețele standard între canale și unitățile de control, precum și compatibilitatea tuturor modelelor la nivel de limbaj-mașină contribuie la realizarea unor configurații flexibile și ușor extensibile. SYSTEM/360 se distinge prin următoarele elemente : un proiect orientat către memorii cu o capacitate foarte mare ierarhizate în funcție de timpul de acces, o gamă largă de funcții de manipulare, tratarea uniformă a funcțiilor de intrare/ieșire, fapt ce avantajează dialogul cu

echipamentele periferice etc. Structura globală se pretează la încorporarea compatibilității programelor pentru un număr mare de mașini aflate pe diferite niveluri de performanță.

Sistemul, proiectat pentru operarea cu un program supervizor, are facilități pentru protecția memoriei, realocarea programului, operarea non-stop și întreruperea programului.

Programul supervizor planifică și guvernează execuția programelor multiple, manevrează condițiile excepționale și coordonează instrucțiunile de intrare/ieșire. Fiabilitatea este sporită prin suplimentarea numărului de componente pentru verificare și diagnoză.

Schema-bloc a calculatorului SYSTEM/360 este prezentată în fig. 2.8.

Comunicația directă cu un număr mare de terminale de viteză mică și cu alte dispozitive de intrare/ieșire se realizează prin intermediul unui canal multiplexor. Comunicația cu dispozitive I/E de mare viteză se face printr-un canal selector. Conceptual, sistemul I/E acționează ca un set de subcanale care operează în mod concurent cu un alt sistem și cu unitatea de procesare.

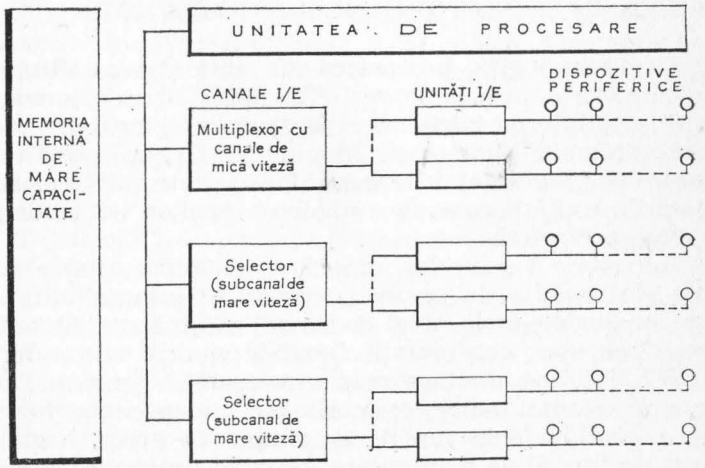



*Fig. 2.8.* Schema-bloc a calculatorului IBM SYSTEM/360.

Fiecare subcanal, dirijat de propriul microprogram, poate guverna operația de transfer a datelor între memorie și dispozitivul de I/E selectat. Un canal multiplexor poate funcționa cu unul sau mai multe subcanale, iar un canal selector funcționează ca un singur subcanal. Unitatea de control a fiecărui dispozitiv I/E este conectată la canale prin intermediul interfețelor standard.

Unitatea de procesare are 16 registre generale de 32 biți folosite pentru adresare, indexare și acumulare.

Opțional, sistemul dispune de patru acumulatoare pentru virgulă mobilă de 64 biți. Includerea registrelor multiple permite folosirea unor memorii mici de mare viteză. Există patru tipuri de procesare : manipularea logică a biților, a șirurilor de caractere și a cuvintelor de lungime fixă ; aritmetica zecimală ; calculul în virgulă fixă ; calculul în virgulă mobilă.

Unitatea de procesare împreună cu controlul central al funcțiilor poartă numele de *unitate centrală de procesare*. Datorită utilizării setului extins de instrucțiuni în SYSTEM/360 controlul este mult mai complex decît în alte calculatoare. Funcțiile de control includ secvențierea fiecărei operații, guvernarea transferurilor de I/E etc.

Dispozitivele de I/E cuprind cititoare de cartele, benzi magnetice ,discuri magnetice, terminale de afișare etc. Aceste dispozitive sînt legate la unitatea centrală prin intermediul unităților de control.

O unitate de control funcționează numai cu acele dispozitive de intrare/ieșire pentru care a fost realizată. Scopul principal urmărit de proiectanții calculatorului SYSTEM/360 a fost realizarea unui dispozitiv extrem de puternic și totodată flexibil pentru păstrarea compatibilității la nivel de program a tuturor membrilor familiei.

## 2.2. STRUCTURI VLSI

Cercetările în domeniul tehnologiei semiconductoarelor au vizat de-a lungul anilor creșterea numărului de biți de memorie pe o structură. Unitățile de control au fost în general construite cu circuite integrate SSI și

MSI. În momentul în care densitatea de componente pe o structură s-a mărit considerabil, pentru realizarea circuitelor LSI au fost create condițiile apariției microcalculatoarelor și în general, a unor calculatoare mult mai compacte.

Primul set de circuite integrate pentru un microcalculator a apărut în anul 1971. Denumit MCS 4, el era alcătuit din următoarele circuite : procesorul Intel 4004 pe patru biți cu 45 de instrucțiuni, memoria RAM 4001, memoria RAM 4002 de 32 biți și Intel 4003 care îndeplinea funcția de adaptor de intrare/ieșire. Procesorul era alcătuit din 3 200 de tranzistoare. Datorită numărului limitat de componente microcalculatoarele prezentau cîteva deficiențe — numărul limitat de coduri de operare, numărul mic de moduri de adresare, viteza redusă a procesorului, posibilitatea ca două sau trei procesoare să adreseze aceeași memorie.

Între timp numărul de componente pe structură a început să crească, iar arhitectura microprocesoarelor a devenit importantă pentru tot mai multe calculatoare. În tabelul 2.4 sînt prezentate cîteva caracteristici ale celor mai importante microprocesoare. Odată cu trecerea ani-

TABELUL 2.4 **Tipuri de microprocesoare**

| | Intel 4004 | Intel 8008 | Moto-rola 6800 | Intel | Zilog Z-80 | Intel 8086 | Moto-rola 6800 | Zilog Z-8000 |
|---|---|---|---|---|---|---|---|---|
| Tehnologie | PMOS | PMOS | NMOS | NMOS | NMOS | HMOS | HMOS | NMOS |
| Număr de pini/capsulă | 16 | 18 | 40 | 40 | 40 | 40 | 64 | 48 |
| Timpul instrucțiunii (μs) | 12,5 | 7,5 | 2 | 2 | 1 | 0,25 | 0,25 | 0,75 |
| Lungimea cuvîntului (biți) | 4 | 8 | 8 | 8 | 8 | 8 | 16 | 16 |
| Memorie maximă (octeți) | 4K | 16K | 64K | 64K | 64K | 1M | 16M | 8M |
| Memoria instrucțiunii (octeți) | 1—2 | 1—3 | 1—3 | 1—3 | 1—3 | 1—4 | 2—6 | 2—10 |
| Setul de instrucțiuni | 45 | 66 | 72 | 111 | 89 | 133 | 59 | 110 |
| Numărul modurilor de adresare | — | 4 | 7 | 5 | 10 | 24 | 10 | 8 |
| Niveluri de întreruperi | 4 | 8 | 1 | 8 | 128 | 256 | 8 | 3 |
| Anul apariției | 1971 | 1972 | 1974 | 1974 | 1975 | 1978 | 1980 | 1979 |



lor arhitecturile microprocesoarelor au înregistrat o evoluție rapidă — de la microprocesoare implementate în capsule cu un număr sporit de terminale, la microprocesoare tot mai complexe (8086, 68000, Z 8000 etc.).

În paralel lungimea cuvintelor a crescut de la 8 la 16 și 32 biți, posibilitatea de adresare de la 64 Kocteți la 1 Moctet și mai mult, modurile de adresare au devenit tot mai complexe, iar numărul structurilor de intrare s-a mărit continuu. Înainte de apariția unor noi seturi de instrucțiuni au fost dezvoltate sisteme de operare și software aplicativ prin intermediul seturilor de instrucțiuni anterioare.

Într-o scurtă dar densă perioadă — de la lansarea pe piață a primului microprocesor s-au scurs numai 15 ani— complexitatea circuitelor integrate a crescut de 100 de ori, iar prețul a scăzut aproximativ în același raport, microprocesoarele revoluționînd conceptul de proiectare în domeniul calculatoarelor și pătrunzînd în viața noastră de zi cu zi.

Performanțele fiecărui nou produs depinde de mai mulți factori : inovațiile din domeniul semiconductoarelor, îmbunătățirea arhitecturilor, design-ul circuitelor, complexitatea software-ului, precum și păstrarea compatibilității cel puțin la nivel de program între membrii aceleiași familii de microprocesoare.

La început au fost realizate așa cum am arătat, circuite integrate simple ; apoi, treptat, s-au acumulat cunoștințele necesare producerii de circuite integrate pe scară largă (LSI — *L*arge *S*cale *I*ntegration), deschizîndu-se astfel calea pentru apariția circuitelor integrate pe scară foarte largă (VLSI).

Deși tehnologiile software se află pe primul plan al eforturilor vizînd realizarea unei noi generații de calculatoare, totuși, circuitele VLSI vor constitui baza ce va permite aplicarea software-ului respectiv. Momentan nu sînt disponibile nici dispozitivele de proiectare și nici tehnologia de fabricare a unor astfel de circuite integrate. Cu toate acestea, în domeniul proiectării circuitelor VLSI s-a realizat un anumit progres prin implementarea pe calculator a metodei de integrare automată.

În prezent au fost deja realizate cîteva circuite integrate care permit construirea calculatoarelor din noua generație. Între aceste circuite se remarcă acceleratorul cu funcționare în virgulă mobilă realizat de firma DEC, *D*igital *E*quipment *C*orporation, precum și unitatea centrală pe 32 biți realizată de firma Hewlett-Packard. Primul circuit conține 125 000 componente pe o structură, iar al doilea 450 000 tranzistoare pe o structură. Se apreciază că circuitele respective constituie realizări premergătoare circuitului integrat cu 1 milion de componente pe structură.

Dezvoltarea rapidă a noii generații de calculatoare va fi influențată și de puterea mare de calcul ce poate fi obținută, prin conectarea în paralel a mai multor circuite VLSI.

După aprecierile specialiștilor, calculatoarele din generația a cincea vor fi construite cu circuite integrate care vor avea fiecare mai mult de 1 milion de componente. Proiectarea unor asemenea circuite complexe trebuie să aibă la bază o gamă largă de informații, incluzînd specificații funcționale, machete, descrieri electrice și date pentru testare.

Odată cu introducerea în fabricație a memoriei DRAM (*D*ynamic *R*andom *A*cces *M*emory) de 4 096 biți — eveniment care a avut loc în anul 1970 — a devenit posibilă realizarea unor circuite LSI și VLSI care să întrunească un număr tot mai mare de caracteristici.

Se estimează că dimensiunile la care s-a ajuns 8—10μm) nu reprezintă o limită, noile tehnologii — litografie cu fascicul de electroni, cu raze X și gravarea cu fascicul de ioni — deschizînd noi căi de investigare în acest domeniu.

Studiile asupra reducerii dimensiunilor circuitelor integrate subliniază avantajele integrării pentru dispozitivele individuale la o scară constantă a integrării. Creșterea scării de integrare este influențată de constanta de timp a interconexiunilor, element care limitează performanțele circuitelor VLSI. Suprafața logicii circuitului VLSI este limitată de densitatea posibilă de realizat a interconexiunilor. În cazul circuitelor de memorie în special interconexiunile limitează semnificativ densita-



tea. Așadar, tehnologia de realizare a interconectării componentelor este cea mai serioasă dificultate întîmpinată în realizarea circuitelor VLSI de mare viteză. Pentru înlăturarea ei, în locul siliciului policristalin sînt folosite siliciuri metalice pentru cel puțin un nivel de interconectare. Aceste siliciuri au o rezistivitate electrică mult mai mică decît siliciul, avînd însă multe din caracteristicile pozitive ale siliciului policristalin, cum ar fi realizarea unui contact electric foarte bun la un singur cristal de siliciu și capacitatea de a forma un oxid de calitate înaltă. Constantele de timp ale unei astfel de interconexiune sînt micșorate prin reducerea rezistenței lor.

Un al doilea domeniu spre care și-au îndreptat eforturile specialiștii a fost creșterea numărului de niveluri de interconectare pentru a permite liniilor individuale să fie mai scurte, fapt ce a condus la apariția unor rezistențe și capacități mai mici.

De menționat că în prezent, cele mai avansate circuite VLSI dispun în mod curent de două sau trei niveluri de interconectare încorporate într-o structură.

## 2.3. ARHITECTURI AVANSATE DE CALCULATOARE

Cu toate diferențele mari existente între structura hardware și performanță, primele patru generații de calculatoare se bazau toate pe un singur model — procesorul realizat de von Neumann. Aceste sisteme sînt secvențiale, efectuînd o singură operație la un moment dat. În plus ele utilizează un singur element de procesare, au o unitate secvențială de control centralizat, posedă un limbaj secvențial de nivel mic, adresare liniară și memorie fixă. Pentru mărirea vitezei într-un sistem secvențial, blocurile din compunerea sa trebuie să funcționeze rapid. Generația a cincea de calculatoare va trebui să asigure viteze de prelucrare mult mai mari decît cele pe care le oferă arhitecturile de tip von Neumann.

Schimbări majore vor afecta limbajele de programare, sistemele de operare și arhitecturile. Deși s-au depus eforturi substanțiale pentru dezvoltarea acestor sisteme

de calcul, oamenii de știință nu și-au spus încă ultimul cuvînt. Fiecare proiect se concentrează asupra unui ansamblu dintr-un sistem complex — compilatoare, dispozitive de intrare-ieșire etc. —, fiind elaborate mai multe variante de soluții. Este probabil ca aceste sisteme — capabile să atingă viteze foarte mari — să fie dezvoltate pînă cînd ideile din domeniul procesării în paralel, materializate acum în marile laboratoare, vor fi implementate în cele mai avansate calculatoare de tip multiprocesor.

În ceea ce privește arhitecturile pentru calculatoarele concurente, cele mai importante aspecte sînt legate de efectuarea controlului, dimensiunea procesorului de bază, comunicațiile topologice, limbajele de programare și alocarea task-urilor.

Primul aspect și poate cel mai important al realizării unei arhitecturi concurente se referă la tipul controlului — centralizat sau descentralizat.

Datorită lungimii globale a cablajului sistemele cu control centralizat trebuie să facă față aspectelor legate de asigurarea unei viteze de lucru corespunzătoare, în timp ce sistemele cu control descentralizat trebuie să realizeze o mai bună coordonare a resurselor sistemului.

### 2.3.1. ARHITECTURILE RISC — O ALTERANTIVĂ A PROCESĂRII PARALELE

Arhitecturile paralele și RISC (*R*educed *I*nstruction *S*et *C*omputer — calculatorul cu set redus de instrucțiuni) reprezintă două modalități de realizare a unor calculatoare cu performanțe ridicate. În unele aplicații proiectele RISC pot elimina arhitecturile paralele sau sistemele multiprocesor.

Un proiect RISC înseamnă mai puține porți și, pentru oricare nivel dat al performanței, mai puțin hardware, consum mic de putere, fiabilitate ridicată și costuri reduse. Timpul de proiectare este de asemenea mai scurt decît pentru un calculator echivalent CISC (*C*omplex *I*nstruction *S*et *C*omputer — calculator cu set complex de instrucțiuni).



Datorită unității centrale care are un set de instrucțiuni redus la cîteva operații (cele care sînt folosite în majoritatea cazurilor) există mai puține instrucțiuni de decodificat și un ciclu mai rapid. De exemplu, este posibil să se ruleze într-un singur ciclu aproximativ 60% din instrucțiunile sistemului. În pofida denumirii — calculator cu set redus de instrucțiuni —, acesta poate furniza multe din caracteristicile mașinilor CISC — memorie, virgulă mobilă, spațiu mărit de adresare și calcule în virgulă mobilă.

Performanțele sistemului CISC pot fi sporite prin mai multe mijloace. Cel mai evident mijloc constă în creșterea vitezei procesării seriale prin utilizarea tehnologiei ECL pentru circuitele integrate sau prin perfecționarea caracteristicilor noilor arhitecturi. Proiectanții de arhitecturi CISC adoptă decizia corespunzătoare analizînd un număr important de factori — fiabilitate, performanța, timpul de proiectare, costul produsului etc.

Odată cu proiectarea sistemului CISC, orientarea către tehnologii mai rapide ori adăugarea unor caracteristici noi, poate fi situată dincolo de limitele tehnologiei actuale. În mod firesc apare tentația reîntoarcerii spre unitățile centrale multiple și la alte forme de arhitecturi paralele. Un proiect RISC, pe de altă parte, simplifică procesul de proiectare și facilitează trecerea spre nivelurile înalte ale performanței. Este posibil să existe o arhitectură RISC care să opereze cu o viteză dublă față de una CISC, ambele sisteme fiind realizate cu microprocesoare.

Mult mai important pentru RISC este faptul că viteza poate fi mărită fără creșterea costurilor de producție, deoarece tehnologia TTL este satisfăcătoare. Unitățile centrale adiționale dintr-un sistem multiprocesor implică creșterea cheltuielilor necesare pentru realizarea sa, datorită numărului mare de suporturi pentru circuite imprimate și sistemului de ventilare.

În principiu sistemele multiprocesor oferă un nivel de fiabilitate mai ridicat decît sistemele cu un singur procesor. Dacă un procesor se defectează, un altul îi ia locul atît din punct de vedere al hardware-ului, cît și al software-ului. Totuși sistemele RISC sînt mai fiabile

decît calculatoarele complexe, datorită arhitecturii lor mai simple, utilizării unui hardware redus, nivelului scăzut al consumului și al disipării de energie, precum și datorită faptului că nu necesită amenajări speciale pentru locul de instalare.

În final se ridică problema determinării performanței unui calculator-serie. Pentru sistemele care structurează o aplicație în părți independente sau în procese multiple, multi-tasking etc., procesele multiple implică cu certitudine creșterea performanței.

Dar pentru un singur program care necesită procesare serială, adăugarea a două sau trei unități centrale nu va duce la îmbunătățirea performanței. Mai mult chiar, va fi influențată viteza de prelucrare a unui singur job. Pe scurt, pentru cîteva aplicații, creșterea procesării seriale prin arhitecturile RISC nu numai că păstrează continuitatea sistemului de operare, ci în mod real determină mărirea vitezei de procesare.

## 2.3.2. PROCESAREA DE TIP PARALEL

Formele de procesare în paralel care implică creșterea vitezei de prelucrare a informației în sistemele electronice de calcul se caracterizează printr-o mare diversitate. Ele pot fi gruptate — nu exclusiv — în cinci clase : suprapunerea (pipelining), unitățile multiple de execuție, vectorizarea, multiprocesarea și concurența.

Suprapunerea — cea mai simplă formă de procesare în paralel — constă în accesarea unui singur cod serial, putînd fi întîlnită la majoritatea superminicalculatoarelor. Prin suprapunere, executarea unei instrucțiuni constă într-o succesiune de faze : aducerea instrucțiunii, decodificarea, calcularea adresei operandului, aducerea operandului și executarea operației. Dacă fiecare din aceste faze poate fi controlată (independent) în calculator, la un moment dat este posibil ca la fiecare nivel al suprapunerii să existe cîteva instrucțiuni în execuție. Adesea, mărirea performanței peste ceea ce rezultă prin suprapunerea fazelor se obține cu ajutorul unităților multiple de execuție (procese dedicate pentru anumite



activități care permit o creștere substanțială a vitezei de execuție a instrucțiunilor). De exemplu, operațiile executate de un procesor în virgulă mobilă se pot suprapune peste cele executate de procesorul de bază.

Utilizarea efectivă a paralelismului în unitățile multiple de execuție este destul de complexă, multe programe fiind concentrate într-o singură declarație.

Spre deosebire de tehnicile menționate, vectorizarea — operație mai complexă decît procedeele de mai sus — nu este transparentă pentru software. De fapt vectorizarea presupune proiectarea unui compilator special. Pînă de curînd operațiile cu vectori erau posibile numai pe supercalculatoare sau în procesoarele matriceale pentru superminicalculatoare și minicalculatoare. În prezent operațiile vectoriale sînt adăugate la arhitecturile de superminicalculatoare. Se poate aprecia că aceste sisteme au performanțe echivalente cu cele mai bune superminicalculatoare cu procesoare vectoriale integrate. Elaborarea unor asemenea sisteme a devenit posibilă întrucît, pe de o parte, tehnologia de realizare a compilatoarelor a evoluat către punctul în care acestea pot transforma programele FORTRAN pentru mașini vectoriale, iar pe de altă parte, densitatea ridicată a circuitelor LSI obișnuite permite integrarea posibilităților vectoriale de prelucrare în arhitecturile de superminicalculatoare chiar de la începutul procesului de proiectare.

Există mai multe forme de multiprocesare, dar în general arhitecturile de acest tip constau din cîteva calculatoare autonome interconectate printr-o memorie globală comună. Memoria respectivă poate fi cuplată cu unitatea de bază de sincronizare a comunicațiilor interprocesor.

Un sistem de multiprocesare este capabil de a executa procesarea interactivă atunci cînd procesele generale au durate scurte de rulare. Sistemele de multiprocesare pot servi simultan mai mulți utilizatori atunci cînd, cel puțin conceptual, procesoarele sînt alocate corespunzător. Uneori, folosirea acestor sisteme în activități complexe, tehnologice și științifice nu este eficientă. Astfel, programele trebuie reproiectate sau restructurate în pro-

cese individuale de dimensiuni mai reduse pentru a putea fi rulate pe un sistem multiprocesor.

În măsura în care limbajul FORTRAN și aplicațiile tehnico-științifice sînt strîns legate, tehnicile menționate nu sînt suficiente. Aceste tehnici nu adresează bucle din programul FORTRAN care nu pot fi vectorizate și nici subrutine din interiorul buclelor sau bucle cu sfîrșit condiționat. Sistemele devin disponibile dacă pot manipula aceste limitări în circulația informației, tratîndu-le în paralel prin forme variate de concurență.

Prin concurență compilatoarele prelucrează în paralel părți importante din program. Unitățile de control care permit procesoarelor-paralel să opereze unde nu o puteau face înainte minimizează sau elimină limitările provocate de vectorizarea programelor FORTRAN. Concurența cere o nouă tehnologie de compilare. De exemplu, dacă compilatorul nu poate vectoriza o buclă, atunci el trebuie să determine dacă codul poate fi rulat în șir continuu pe cîteva procesoare în același timp. Acest cod concurent este dependent de datele prelucrate și procesorul trebuie astfel sincronizat încît programul să fie executat la fel ca pe o mașină serială.

Procesarea paralelă permite executarea mai rapidă a programelor. Problemele complexe din știință și tehnologie, a căror rezolvare necesită o mare putere de calcul ce nu poate fi asigurată de calculatoarele convenționale, vor fi soluționate în curînd cu ajutorul noilor arhitecturi paralele și al software-ul procesat în paralel. Menționăm că de aproape două decenii sînt întreprinse cercetări pentru realizarea calculatoarelor cu procesare paralelă. La îndeplinirea acestui deziderat au contribuit doi factori ; elaborarea de microprocesoare pe 32 de biți la un preț redus și nevoile tot mai mari de putere de calcul.

Arhitectura calculatoarelor tradiționale a și atins limitele sale de dezvoltare. Noile sisteme cu procesare paralelă și-au găsit utilizarea nu numai pentru rezolvarea problemelor științifice și tehnologice actuale, ci și pentru soluționarea acelor probleme considerate aproape inabordabile.



Comunicarea între procesoare este posibilă prin împărțirea memoriei comune sau prin combinarea tehnicilor de comunicație. Același sistem de operare poate fi instalat pe fiecare procesor.

Sisteme de operare diferite pot fi prezente pe fiecare procesor, iar, într-o altă alternativă, numai părți din sistemul de operare pot fi rezidente pe un procesor particular.

Un task specific dintr-un program de aplicație poate fi „spart" în părți mai mici de către un compilator pentru a fi executat de mai multe procesoare ; pe de altă parte, fiecare procesor poate manipula un anumit task. Această împărțire a task-urilor conduce la divizarea calculatoarelor în mai multe categorii dependente de numărul de subprograme în care este divizat un program sau de numărul procesoarelor care lucrează la o problemă.

Mașinile care procesează în paralel pot fi angrenate la rezolvarea algoritmilor existenți și a software-ului aplicativ existent sau pot fi proiectate special pentru algoritmi și programe noi. Pe scurt, există o mulțime de suprapuneri și clasificări asupra cărora se poate discuta. Într-adevăr, adesea software-ul (compilatoarele, algoritmii, controlul curent și programele aplicative) este cel care, datorită procesării în paralel, determină eficiența calculatoarelor de acest tip. În timp ce multe arhitecturi cu procesare în paralel erau construite, cele cu matrice sistolice se aflau doar în stadiul de proiect.

Calculatoarele complexe necesită, cel mai adesea, sisteme de operare dedicate sau limbaje de programare pe care puțini utilizatori potențiali le cunosc. Drept rezultat, puține sau aproape nici una din aceste mașini nu ajung în faza de producție de serie. Cu toate acestea cîteva produse proiectate de specialiști în domeniul sistemelor electronice de calcul sînt acum pregătite pentru a intra în producția de serie, în scopul înlocuirii treptate a calculatoarelor tradiționale.

Fiecare proiectant a ales o arhitectură pe care a crezut-o mai potrivită pentru o anumită aplicație. Referitor la diferențele arhitecturale, aceste sisteme multiprocesor pot fi clasificate aproximativ în sisteme cu procesare dis-

tribuită sau cu procesare paralelă ; rețele locale ori cu flux de date, rețele comutate sau sistolice etc. Clasificarea menționată prezintă unele suprapuneri ; totodată puține arhitecturi pot fi considerate pe deplin originale. De exemplu, cele trei forme de paralelism — flux de control, flux de date și flux de rezultate — care domină astăzi activitatea de proiectare a calculatoarelor cu procesare în paralel sînt în mod intim legate de aspectele referitoare la organizarea instrucțiunilor în cadrul programului.

Pentru numeroase aplicații, folosirea sistemelor de calcul cu procesare în paralel vizează viteza de prelucrare a informației. Rezultatele obținute în această privință sînt dependente de cîțiva factori, dintre care menționăm : natura aplicației ; alegerea arhitecturii hardware, precum și a sistemului de operare care controlează comunicația interprocesor și sincronizarea ; inteligența algoritmilor paraleli ; modul în care software-ul este orientat către arhitecturile paralele.

În conformitate cu unul dintre punctele de vedere asupra procesării în paralel, un sistem care folosește $N$ procesoare și rulează o aplicație împărțită în $N$ taskuri ar trebui să respecte o relație liniară între performanță și numărul de procesoare. Cu alte cuvinte, un sistem cu $N$ procesoare poate fi de N ori mai rapid decît un singur procesor. Un alt punct de vedere, expus de Marvin Minsky de la MIT (*Massachusetts Institute of Technology*) consideră că pentru $N$ foarte mare, viteza este proporțională cu $\log_2 N$.

O părere mai optimistă este susținută de Eugene Amdahl, care a emis legea ce-i poartă numele. Această lege statuează că dacă un calculator are două viteze de operare (una pentru coduri paralele ce pot fi executate rapid și alta pentru coduri secvențiale ce nu pot fi executate cu viteză mare) viteza mai mică domină performanțele. Rezultă o altă relație între viteză și numărul de procesoare și anume $N/\log N$. Deși această relație este adesea adevărată, ea poate fi susceptibilă de îmbunătățiri. Procedînd în consecință, un cercetător a dezvoltat tehnica de programare care să transforme programul



FORTRAN existent într-un program ce poate fi rulat pe un calculator de tip paralel cu o viteză de 0,3 $N$.

Unii oameni de știință au ajuns la concluzia potrivit căreia controlul concurent, bazat pe hardware dedicat, combinat cu un compilator pentru generarea de coduri paralele din programele existente permite atingerea performanțelor teoretice (ansamblul de $N$ procesoare fiind de $N$ ori mai rapid decît un procesor) pentru cîteva aplicații. Folosind o arhitectură LSI concurentă similară cu cea a sistemului Cosmic Cube lansat de firma Intel, cercetătorii de la Institutul de Tehnologie din California au reușit să atingă viteze de 0,94 $N$ pentru sisteme cu 64 și 128 de procesoare.

Legea lui Amdahl poate fi aplicată la situații comune în care anumite coduri sînt executate de către procesoarele paralele, iar altele sînt executate de către un singur procesor secvențial.

Creșterile vitezei într-un sistem cu $N$ procesoare ocupă o plajă de valori cuprinse între viteza cea mai mare, atinsă în cazul codurilor paralele, și viteza mică obținută în cazul codurilor seriale, fiind dependente de numărul acestor coduri. Creșterea vitezei de prelucrare echivalentă pentru $N$ procesoare, comparativ cu un singur procesor, se poate aprecia cu relația $1/F+(1-F)/N$, unde $F$ reprezintă proporția de coduri seriale. Cîștigul posibil ce poate fi obținut prin procesarea în paralel este complet limitat dacă există un număr mare de coduri seriale.

Este de la sine înțeles că îmbunătățirile semnificative ale vitezei necesită un software paralel foarte evoluat. Nu este totuși destul de clar, uneori, cîți algoritmi folosiți în mod curent pentru un singur procesor au un paralelism inerent sau cîți algoritmi și software mai sînt necesari.

Aceste estimări se bazează pe presupunerea că instrucțiunile executate pe procesoarele paralele sînt manipulate în același mod ca și cele de pe un singur procesor secvențial.

Sistemele multiprocesor necesită instrucțiuni suplimentare pentru sincronizarea procesului, deși comunicațiile interprocesor și algoritmii paraleli pot necesita noi

instrucțiuni. Puterea de calcul eficientă presupune că toate procesoarele lucrează permanent. Din păcate toate problemele care apar la interconectarea mai multor procesoare nu sînt încă înțelese în totalitate.

Procesarea semnalelor digitale reprezintă un exemplu care demonstrează că avantajele procesării în paralel sînt potențiale. Progresele recente în domeniile circuitelor VLSI, software-ului, aplicațiilor etc., au dus la creșterea intensivă a puterii de calcul în multe domenii — analiza imaginilor, recunoașterea vorbirii, analiza spectrală ș.a.

Procesarea în paralel constituie un element important care determină îmbunătățirea prelucrării semnalelor digitale. În timp ce aplicațiile procesării semnalelor digitale reprezintă o importantă premiză pentru folosirea arhitecturilor de procesare în paralel, dimensiunile paralelismului în implementarea sistemelor-expert este mai puțin clară. Una din dificultățile cu care se confruntă sistemele-expert bazate pe LISP constă în aceea că programele mari, cu sute și adesea mii de reguli sau clauze, se execută lent chiar pe calculatoarele dedicate. Mai mult, pentru sistemele-expert software-ul este prevăzut să permită atingerea vitezei de un miliard de LIPS (*L*ogical *I*nferences *P*er *S*econd), limita la care nu se poate ajunge cu hardware-ul de astăzi. Procesarea în paralel poate fi o soluție pentru mărirea vitezei de execuție din sistemele-expert. Un asemenea sistem folosește, în principiu, una sau două strategii de control ori o combinație a acestora, pentru a găsi soluția unei probleme. În așa-numita strategie forward-chaining — o combinație între limbajul de inteligență artificială, un sistem de dezvoltare și un sistem-expert — datele inițiale ale unei probleme reprezintă punctul de la care se pornește pentru stabilirea inferențelor logice.

Noile date sînt deduse pînă la desprinderea concluziilor ; în acest timp software-ul construiește baza sa de cunoștințe, utilizată din acel moment pentru expertiză.

În contrast, sistemele în strategie backward-chaining pornesc de la un anumit scop și lucrează pentru a găsi



o secvență de reguli sau interfețe care pot să descrie scopul respectiv funcție de baza de cunoștințe existente. De menționat că unele sisteme-expert combină strategiile prezentate. După estimările mai multor cercetători în domeniul sistemelor-expert se așteaptă ca spre sfîrșitul actualului deceniu un calculator din generația a cincea să posede o rată de execuție cuprinsă între 50 de milioane și un miliard de LIPS. Precizăm că, în prezent, un calculator execută între 30 și 50 LIPS. Se apreciază totodată că, în cazul sistemelor-expert, procesarea paralelă trebuie investigată amănunțit la nivelul regulilor cercetării și limbajului, dar nu este încă destul de clar cum se aplică paralelismul la nivelurile respective.

După aprecierea unor specialiști, calculatorul cu set redus de instrucțiuni (RISC) poate lucra la fel de bine ca un procesor paralel.

Bazîndu-se pe rezultatele obținute pînă în prezent, mulți observatori se îndoiesc de faptul că în sistemele-expert există destul paralelism inițial pentru utilizarea eficientă a marilor arhitecturi de procesare în paralel.

Fluxul secvențial de tratare a informației, întîrzierile și adesea structura de bază a programării logice limitează cîștigul ce poate fi obținut prin utilizarea paralelismului.

În prezent se depun eforturi susținute pentru realizarea proiectelor noilor modele bazate pe prelucrarea în paralel. Unele firme specializate în producția de calculatoare de mare performanță au început operațiile specifice pentru elaborarea unor sisteme tehnice de calcul foarte complexe — modele operaționale de procesoare sistolice, arhitecturi paralele de tip MIMD, așa-numitele ultracalculatoare care utilizează tehnologia MIMD bazată pe microprocesoarele multiple 68010, calculatorul COSMIC CUBE care conectează pînă la 128 de procesoare în paralel etc. O utilizare importantă a echipamentului COSMIC CUBE constă în simularea zborurilor spațiale. Simularea spațială, la fel de bine ca alte tipuri de simulare, va reprezenta o aplicație tehnico-științifică fundamentală pentru numeroase sisteme electronice de calcul cu procesare în paralel.

În domenii de activitate dintre cele mai diverse — proiectarea circuitelor integrate, proiectarea aripilor de avion, prognozele meteo etc. — complexitatea sarcinilor a crescut atît de mult încît simularea pe calculator este singura modalitate de a găsi soluții eficiente.

Firmele producătoare de calculatoare și-au intensificat eforturile pentru realizarea de sisteme cu procesare în paralel. Printre preocupările în acest domeniu — unele finalizate, iar altele aflate în stadiul de proiect — menționăm o versiune a supercalculatoarelor produse de firma Cray Research cu pînă la patru procesoare lucrînd în paralel, proiectul calculatorului paralel CEDAR (scopul urmărit de CEDAR este realizarea sistemelor paralele cu zeci, sute și chiar mii de procesoare), proiectul dispozitivului cu zece procesoare sistolice utilizînd circuite integrate care se află pe piață (matricea utilizează o conexiune de tip multibus la un procesor 68010 ; totodată procesorul sistolic matriceal folosește procesorul 68020 și bus-ul VME) etc.

Pentru utilizatorii din domeniul științei și tehnologiei diferențele de opinie asupra gradului de paralelism din orice sistem de calcul particular apar ca un exercițiu de semantică. Nuanțele clasificării procesării în paralel sînt mai puțin importante decît prețul destul de ridicat al calculatoarelor electronice de acest tip.

În privința unei asemenea clasificări trebuie ținut seama de faptul că informațiile de care dispunem pînă în acest moment nu pot oferi o imagine exactă asupra sistemelor electronice de calcul care folosesc procesarea în paralel. Totuși, în fig. 2.9 prezentăm o clasificare primară.

Tipic pentru noile direcții de dezvoltare a arhitecturilor cu procesare în paralel este sistemul Flex/32 Multicomputer. Acest echipament utilizează elemente de procesare multiplă din familia 32032, fiecare lucrînd cu o memorie locală și putînd accesa bus-ul comun al memoriei centrale.

Caracteristic unei versiuni a calculatoarelor este posibilitatea ca într-o singură incintă să fie cuplate mai mult de 20 de elemente de procesare.



Unul din sistemele Flexible Computer se află la Universitatea din Indiana (S.U.A.), unde cercetările în domeniul procesării paralele și, în special, în domeniul rețelelor comutate se află în curs de desfășurare. În

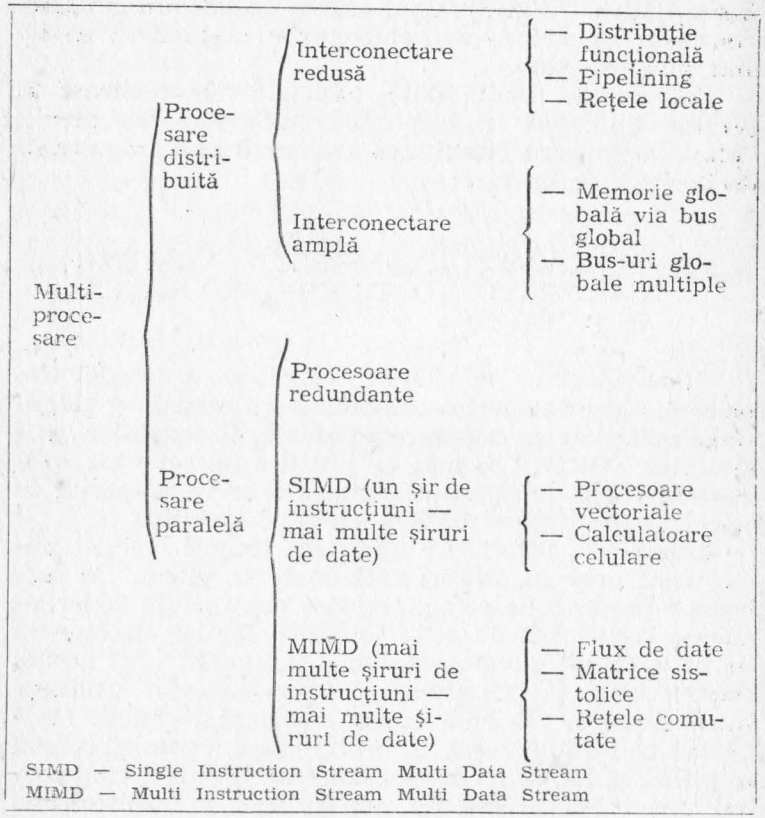

Multi-procesare
Procesare distribuită
Interconectare redusă
— Distribuție funcțională
— Pipelining
— Rețele locale
Interconectare amplă
— Memorie globală via bus global
— Bus-uri globale multiple
Procesare paralelă
Procesoare redundante
SIMD (un șir de instrucțiuni — mai multe șiruri de date)
— Procesoare vectoriale
— Calculatoare celulare
MIMD (mai multe șiruri de instrucțiuni — mai multe șiruri de date)
— Flux de date
— Matrice sistolice
— Rețele comutate
SIMD — Single Instruction Stream Multi Data Stream
MIMD — Multi Instruction Stream Multi Data Stream

*Fig. 2.9.* Tipuri de procesare paralelă.

această instituție se lucrează la realizarea unui dispozitiv compus din mai multe structuri VLSI care pot comuta orice procesor al acestui sistem multiprocesor la oricare din mulțimea de memorii. Această modalitate de comutare reprezintă elementul principal în arhitecturile paralele ; menționăm că o schemă unică de comu-

Pînă în prezent direcțiile specificate mai sus au fost greu accesibile. Totuși compilatoare speciale sînt capabile să vectorizeze cîteva programe ; întrucît ele nu folosesc în mod complet avantajele arhitecturilor de tip paralel aplicațiile lor au fost limitate la problemele de cercetare speciale.

Hardware-ul sistemului Alliant FX/8 (fig. 2.10) lucrează cu programele FORTRAN existente ; el utilizează pînă la opt procesoare de înaltă performanță care lucrează în paralel pentru o singură aplicație, avînd acces la aceeași memorie globală. Hardware-ul sistemului FX/8 folosește microprogramarea, procesoare suprapuse implementate cu circuite de înaltă densitate, denumite elemente de calcul (CE — *C*omputational *E*lements).

Elementele de calcul implementează un set complet de instrucțiuni scalare. Aceste elemente, dispunînd de o unitate de virgulă mobilă și o unitate vectorială, precum și de un hardware care implementează un set de instruc-

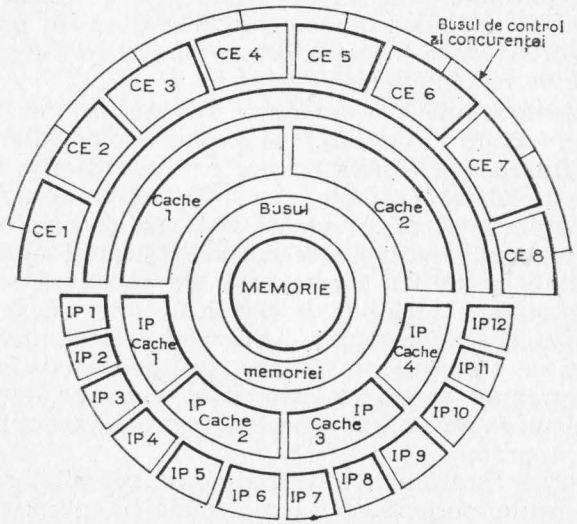



*Fig. 2.10.* Arhitectura sistemului de calcul Alliant-FX/8



tare a fost dezvoltată de firma International Parallel Machines.

Cele mai multe dintre noile produse software sînt de fapt versiuni ale sistemului de operare UNIX cu sau fără îmbunătățiri. Utilizarea unui sistem de operare standard cum este Unix face ca arhitecturile nestandard să fie mult mai acceptate.

Deși rapide, multe din calculatoarele realizate în prezent utilizează în principal multiprocesarea pentru aplicațiile în care fiabilitatea este mult mai importantă decît viteza de operare.

### 2.3.3 SUPERCALCULATOARELE — SISTEME CARE EXTIND OPȚIUNILE PROCESĂRII ÎN PARALEL

Orice calculator nou cu procesare de tip paralel trebuie să întrunească trei caracteristici de bază — asigurarea rulării bazei software existente, în special a programelor FORTRAN, fapt ce justifică aproape tot software-ul folosit în aplicațiile științifice ; viteza sporită de calcul ; eficiența efectuării operațiilor specifice.

Programele FORTRAN existente trebuie rescrise numai dacă apar modificări referitoare la viteză. De asemenea, procesul de compilare care convertește codurile-mașină secvențiale în coduri-mașină paralele trebuie să fie transparent programatorului, indiferent de natura codului-sursă ori de numărul procesoarelor utilizate. Între viteză și eficiență există o strînsă legătură. Dacă un calculator utilizează $N$ procesoare în paralel, viteza ar trebui să fie de $N$ ori mai mare decît atunci cînd același număr de procesoare este utilizat în arhitecturile von Neumann convenționale. Această situație impune reducerea comunicațiilor între procesoare. Astfel va exista un mic cîștig în ceea ce privește viteza de execuție. Din cauza cerinței menționate modalitatea ce trebuie folosită pentru a asigura o procesare paralelă mai eficientă și mai rapidă constă în sincronizarea hardware a procesoarelor centrale cu dispozitivele de intrare/ieșire.

țiuni pentru controlul concurenței, permit programarea virtuală a memoriei. Există două calculatoare de tip Alliant cu procesarea informației în paralel : supercalculatorul FX/8, care utilizează pînă la opt procesoare, și superminicalculatorul FX/1, care folosește un singur procesor.

Hardware-ul necesar controlului concurenței minimizează comunicațiile software, aspect datorat faptului că elementele de calcul au task-uri asignate, elementele respective trebuind să fie sincronizate.

Precizăm că funcțiile respective sînt independente de numărul procesoarelor paralele și de programator. Arhitecturile sistemului FX/8 comportă două forme de procesare în paralel. În primul caz, procesarea în paralel se realizează pe elementele de calcul, fiecare avînd capacitatea de a executa operații în dublă și simplă precizie, ajungînd pînă la 11,8 MFLOP (*M*illion *FL*oating point *OP*erations per second — milioane de operații în virgulă mobilă pe secundă) — performanță de vîrf pentru operațiile vectoriale în virgulă mobilă. Menționăm că pot fi interconectate pînă la opt elemente de calcul (prin intermediul unui bus de control dedicat) cu un procesor al memoriei *cache*, accesat printr-un *switch* ce permite o viteză de 376 Mbytes/s.

A doua formă de procesare de tip paralel constă dintr-un spațiu extensibil al procesoarelor interactive (IP — *I*nteractive *P*rocesor) care execută diferite operații în paralel cu celelalte elemente active din sistem. Procesoarele interactive rulează sistemul de operare (inclusiv toate dispozitivele de intrare/ieșire) și aplicațiile interactive, permițînd elementelor de calcul să se concentreze asupra sarcinilor de calcul.

Pentru operațiile rapide, controlul concurenței (care depinde de transmiterea datelor pe un bus de control al concurenței) se aplică tuturor elementelor de calcul din cadrul complexului de calcul pentru execuția unui singur program.

Datorită controlului concurenței, un compilator FORTRAN poate perfecționa mai mult o aplicație decît atunci cînd aceeași operație este realizată numai cu o arhitectură vectorială. Ca urmare, un sistem FX/8 va



atinge un nivel de performanță mai ridicat decît nivelul realizat pe un procesor vectorial tradițional. De exemplu, buclele DO care se execută în mod vectorial pe un supercalculator sînt compilate pentru a rula ca vectori paraleli pe sistemul FX/8. De asemenea, lucrul în concurență permite procesarea în paralel a mai multor coduri decît în cazul cînd acestea sînt rulate în modul scalar pe calculatoarele vectoriale.

Un element de calcul este proiectat să suporte pînă la cinci instrucțiuni aflate simultan în diferite faze de execuție. Fiecare element de calcul conține cîte o unitate pentru suprapunerea instrucțiunilor și a vectorilor, unitatea de virgulă mobilă și o unitate de control a concurenței.

Unitatea pentru instrucțiuni optimizează executarea instrucțiunilor scalare. Cele cinci niveluri de suprapunere constau dintr-o secțiune de control, un analizor de instrucțiuni, o memorie RAM, un microsecvențiator și 16 Kocteți pentru memoria cache (fig. 2. 11). Aceste unități, procesorul de instrucțiuni și unitatea de translatare a adresei comunică printr-un bus de comandă a instrucțiunilor.

Activitatea analizorului de instrucțiuni prezintă un interes deosebit, întrucît este critică pentru manipularea dependențelor între instrucțiunile comune. Printre alte acțiuni analizorul de instrucțiuni recepționează și codurile operanzilor din fluxul de date, pe care le decodifică pentru generarea adreselor microprogramului.

Cel mai important aspect se referă totuși la memorarea cîmpurilor instrucțiunii pentru codurile de operare aflate în diferite faze de execuție și care verifică dependențele dintre instrucțiuni.

Unitatea de virgulă mobilă și unitatea pentru suprapunerea vectorilor constau din multiplicatoare, sumatoare și divizoare dezvoltate cu circuite VLSI în tehnologie NMOS, precum și dintr-o matrice de porți realizate în tehnologie CMOS cu opt registre vectoriale de 32 de elemente.

Switch-ul elementului de calcul, reprezentînd centrul elementelor de calcul, are rolul de a realiza inter-

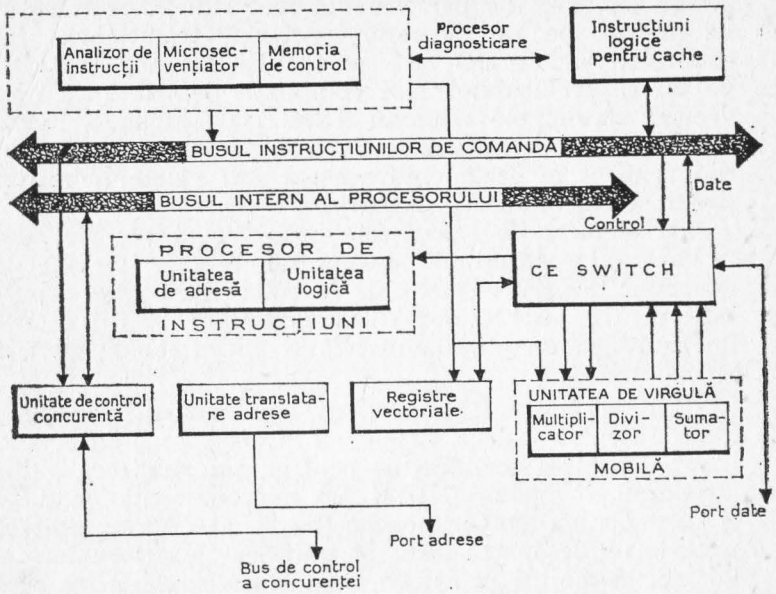



*Fig. 2.11.* Schema-bloc a unui element de calcul (CE).

fața între toate unitățile funcționale ale elementului de calcul și memorie. El asigură și alinierea datelor înspre sau dinspre memorie. Fluxurile multiple de date din elementele de calcul permit realizarea simultană de operații multiple. Unitatea de adresă calculează adresele vectorilor și permite accesul la elementele vectorului, în timp ce unitatea de virgulă mobilă calculează vectorul rezultat.

Unitatea de control a concurenței realizează interfața cu unitatea de instrucțiuni a elementelor de calcul. Elementele de calcul nu pot atinge viteza maximă dacă echipamentul nu este dotat cu un subsistem de memorie rapidă. În cazul sistemului FX/8 memoria principală și memoria cache permit adresarea lor de către toate procesoarele care operează în paralel.

FX/8 conține 12 procesoare interactive (IP) ; fiecare din ele incorporează un procesor 68012 de 12 MHz, o



memorie RAM de 512 Kocteți, 64 Kocteți EPROM, un dispozitiv pentru serializarea și deserializarea informației, interfețe cu memoria cache și cu multibus-ul, bus-ul de adrese, bus-ul de date. Fiecare procesor interactiv accesează memoria globală printr-o arhitectură identică cu a elementului de calcul. În figura 2.12 se prezintă schema-bloc a unui procesor interactiv.

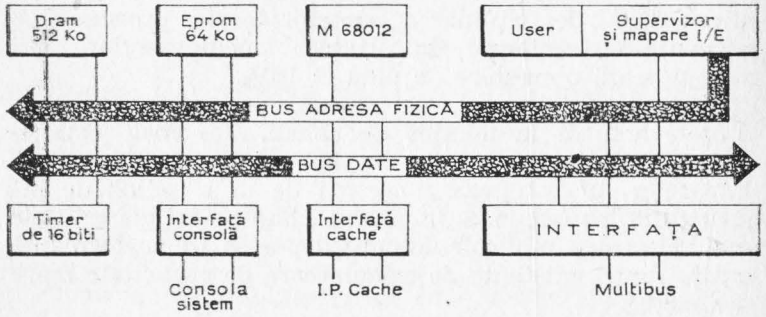



*Fig. 2.12.* Schema-bloc a unui procesor interactiv (IP).

Spațiul de adresă virtuală pentru utilizatori este de 2 Gocteți. Memoria cache a IP — de 128 Kocteți — este împărțită în 4 blocuri, fiecare în valoare de 32 Kocteți. Memoria cache conectează IP și dispozitivele periferice ale acestora la un bus sincron al memoriei (cu o rată de 94 Mocteți/s). Bus-ul memoriei — bidirecțional — are 64 biți pentru date, 8 biți pentru detectarea și corectarea erorilor, 28 de biți de adresă și un număr de semnale de control.

În lipsa intervenției programatorului, lucrul în concurență este singurul element care poate determina creșterea vitezei de execuție a buclelor sau a operațiilor matriceale de un număr de ori apropiat de numărul elementelor de calcul. Vectorizarea permite o creștere adițională de patru ori, astfel încît, în final viteza poate fi de 32 de ori mai mare decît viteza unui CE în modul scalar.

ELXSI 6400 a atras atenția în mod deosebit datorită bus-ului său, care permite ca mai mult de 10 unități de procesare să aducă informații din memorie fără ca viteza de operare să fie afectată în mod deosebit. Bus-ul are o lățime de 64 de biți și permite o rată de transfer a

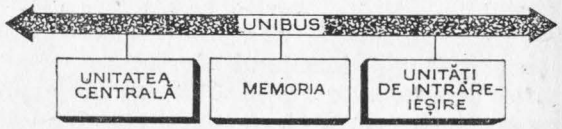



*Fig. 2.13.* Schema-bloc a unui minicalculator DEC.

datelor de 200 milioane de octeți pe secundă. Superminicalculatorul utilizează instrucțiuni și date cu format de lungime variabilă ; formatul datelor include octeți, trei lungimi pentru numere întregi și numere zecimale. Instrucțiunile pot avea lungimi de la 16 biți, pentru operații între registre, pînă la 72 de biți pentru operații registru-memorie.

Sistemul ELXSI poate adresa o memorie internă de 192 megaocteți. Noul sistem VAX 8600 utilizează, de asemenea, pentru creșterea memoriei adresabile un bus extins. Lungimea cuvîntului este de 32 de biți, calculatorul avînd posibilitatea de a aduce deodată din memorie 128 de biți. Astfel, cu un ciclu de memorie relativ lent (560 ns) procesorul poate adresa într-o secundă, pînă la 230 de Mocteți.

În domeniul calculatoarelor de capacitate mare se remarcă echipamentul de tip VP 200 (Fujitsu) și S. 810 (Hitachi). Aceste calculatoare realizează viteze de aproape 400 de milioane operații pe secundă, fiind superioare calculatoarelor de același gen produse de firma IBM care nu ating decît o zecime din viteza menționată mai sus. Calculatorul VP 200 este aproximativ echivalent cu supercalculatorul CRAY XMP-1. Deosebirea constă în faptul că primul lucrează cu vectori de lungime variabilă, iar al doilea, cu vectori de lungime fixă.

Creșterea performanțelor calculatoarelor electronice a afectat și sistemele destinate unor activități speciale, sisteme cunoscute și sub denumirea de procesoare matri-



## 2.4. TENDINȚE ÎN DEZVOLTAREA SISTEMELOR DE CALCUL

Din studiile de prognoză rezultă că pînă la sfîrșitul secolului nostru producția și desfacerea de microcalculatoare, minicalculatoare și calculatoare de capacitate mare vor cunoaște o creștere semnificativă. În domeniul microcalculatoarelor se estimează o creștere cu 40—60%, stimulată și de extinderea rețelelor locale, precum și a software-ului integrat. În privința calculatoarelor mari se estimează o creștere de pînă la 16%.

Mai mic, mai rapid, mai puternic — acestea sînt tendințele actuale în tehnica de calcul. Cea mai semnificativă creștere s-a produs în domeniul superminicalculatoarelor, unde viteza a crescut de la 1 milion de instrucțiuni pe secundă la 4 sau chiar 5 milioane. Cele mai puternice minicalculatoare depășesc în performanță multe tipuri existente de calculatoare de capacitate mare sau medie.

În ceea ce privește calculatoarele de capacitate mare s-au realizat viteze de sute de milioane de instrucțiuni pe secundă. S-au făcut progrese remarcabile în aproape toate domeniile tehnicii de calcul — al supercalculatoarelor convenționale, al procesoarelor matriceale, al arhitecturilor paralele etc.

În domeniul minicalculatoarelor evoluția se axează în principal pe epuizarea tuturor posibilităților tehnologice, dezvoltările constînd în miniaturizare, utilizarea circuitelor LSI și VLSI, creșterea numărului de procesoare, generalizarea procesoarelor de 32 de biți etc.

Noile procesoare realizate sînt comparate, din punctul de vedere al performanțelor, cu produsele firmei DEC și, în special, cu superminicalculatorul VAX 11/780, care realizează ceva mai mult de 1 milion de instrucțiuni pe secundă. În fig. 2.13 este prezentată structura de bază a unui minicalculator DEC.

Procesoarele realizate în ultimul timp asigură viteze de 4—5 milioane de instrucțiuni pe secundă, între acestea remarcîndu-se calculatoarele HARRIS 1000, ELXSI 6400, GOULD 9000 și noul produs DEC-VAX 8600. Calculatorul

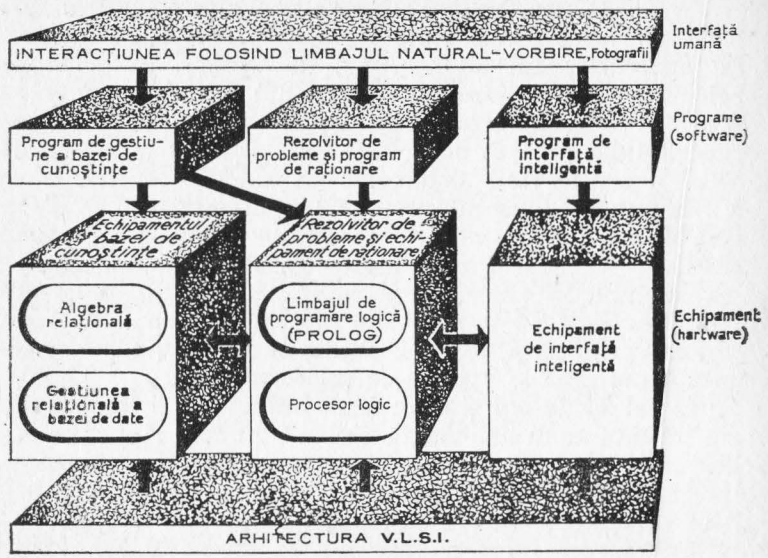

INTERACȚIUNEA FOLOSIND LIMBAJUL NATURAL-VORBIRE, Fotografii
Interfață umană
Program de gestiune a bazei de cunoștințe
Rezolvitor de probleme și program de raționare
Program de interfață inteligentă
Programe (software)
Echipamentul bazei de cunoștințe
Rezolvitor de probleme și echipament de raționare
Algebra relațională
Gestiunea relațională a bazei de date
Limbajul de programare logică (PROLOG)
Procesor logic
Echipament de interfață inteligentă
Echipament (hartware)
ARHITECTURA V.L.S.I.

*Fig. 2.14.* Schema-bloc a unui calculator din generația a cincea.

serie de specialiști consideră că sumele alocate pot fi folosite mai eficient pentru dezvoltarea pe scară largă a supercalculatoarelor utilizate în prezent în aplicațiile științifice.

Denumite în mod obișnuit supercalculatoare și, uneori, calculatoare din generația a patra, aceste echipamente au un hardware foarte evoluat, axat pe creșterea vitezei de calcul și a capacității memoriei, caracteristici îmbunătățite în mod continuu pentru a satisface cererile pieții. Desigur, dezvoltările tehnologice pentru creșterea performanțelor sistemelor de calcul din generația a patra reprezintă un cîștig pentru calculatoarele din următoarea generație.

Indiferent de implicațiile pe care le presupune apariția noii generații de calculatoare, un aspect deloc neglijabil se referă la necesitatea pregătirii unui număr mai mare de specialiști în domeniul sistemelor electronice de calcul.



ceale. Aceste dispozitive au atins viteze foarte mari în prelucrarea datelor numerice. În multe cazuri procesoarele matriceale pot înlocui supercalculatoarele care sînt foarte scumpe.

Sînt cunoscute și alte sisteme electronice de calcul care folosesc procesarea în pararel, sisteme deosebit de performante. Printre acestea se numără calculatorul propus de firma *Scientific Computer Systems*, prevăzut să fie lansat pe piață în cursul acestui an, care va asigura rularea programelor FORTRAN existente. În plus, echipamentul este prevăzut cu un sistem de operare compatibil cu supercalculatorul CRAY, avînd posibilitatea de a rula aplicații dezvoltate pentru această mașină.

Menționăm, de asemenea, echipamentul SYSTEM 6400 oferit de firma ELXSI, care posedă un bus cu o viteză de 320 Mbiți/s ce permite comunicațiile între procesele software, capabile să adreseze un spațiu de 4 Gocteți.

La începutul deceniului actual Japonia anunța un program de lungă durată pentru dezvoltarea calculatoarelor electronice, prin care își propunea extinderea fără precedent a domeniilor acestei științe, astfel încît să se ajungă în situația ca supercalculatorul să devină expert consultant pentru diferite organe de decizie.

Aplicațiile propuse vizau în egală măsură creșterea productivității în domeniile vieții economice, conservarea energiei, expertiza medicală, educație etc.

Acest proiect îndrăzneț a fost denumit *calculatorul din generația a cincea* (fig. 2.14).

Într-adevăr, în timp ce generațiile anterioare de calculatoare pot fi considerate sisteme unitare realizate cu diferite componente corespunzătoare unor faze ale progresului electronicii — tuburi cu vid, tranzistoare, circuite integrate și circuite integrate pe scară largă și foarte largă — generația a cincea ar trebui să fie marcată pe lîngă un hardware realizat cu circuite integrate pe scară ultralargă și de către un software avansat, iar în particular prin utilizarea extensivă a inteligenței artificiale.

Datorită existenței unei anumite incertitudini asupra succesului rapid al programului generației a cincea, o

În scurta dar densa sa istorie prelucrarea informațiilor a afectat aproape toate domeniile societății moderne. Se apreciază, din acest punct de vedere, că cerințele referitoare la folosirea mai eficientă a informațiilor se vor afla la baza tuturor activităților umane. Calculatoarele digitale reflectă o evoluție de aproape o jumătate de secol a arhitecturii sistemelor de calcul secvențiale ce a început în deceniul al patrulea al secolului XX. În cadrul acestei arhitecturi, o unitate centrală de calcul (CPU — *C*entral *P*rocessing *U*nit) manipulează fiecare instrucțiune în parte, acționează asupra ei, păstrează temporar rezultatele sau le stochează în memorie. Acest proces continuă pînă cînd ultima instrucțiune este executată sau pînă la oprirea calculatorului.

În cei 50 de ani s-au realizat îmbunătățiri importante ale unității centrale. Viteza de bază cu care sînt manipulate instrucțiunile a crescut constant datorită numeroaselor îmbunătățiri tehnologice realizate an de an, arhitectura calculatorului secvențial rămînînd însă intactă.

La început, cînd fiecare bit de memorie era foarte prețios, puteau fi memorate și executate numai programe de dimensiuni reduse. Dezvoltarea memoriei *solid-state* și îmbunătățirea continuă a arhitecturilor au permis rularea și executarea unor programe mult mai mari. Se apreciază că la sfîrșitul anilor 80 sistemele convenționale vor fi capabile să execute peste 100 milioane de instrucțiuni pe secundă și să adreseze memorii de ordinul gigaocteților.

În acest punct îmbunătățirile hardware-ului nu mai sînt suficiente pentru a garanta creșterea fără limite a vitezei de calcul pentru tipul de arhitectură analizat. Simpla existență a unui subansamblu hardware care să execute instrucțiuni mai rapid și simplificarea procesului de programare nu garantează că o problemă particulară poate fi rezolvată mai repede.

Sistemele dotate cu software bazat pe inteligență artificială vor putea într-o zi deține controlul asupra configurației sistemului de calcul, generînd automat programe necesare satisfacerii cerințelor imediate. Sistemele noi de operare și programele axate pe aplicații vor favoriza progresul în acest domeniu.



Sistemele care astăzi sînt implementate cu circuite integrate de mare viteză (cu un timp de acces foarte mic) vor putea fi realizate în viitor cu circuite mult mai lente, dar cu o arhitectură complet diferită.

Experiențele arată că sistemele de calcul din următoarea generație vor fi capabile să înțeleagă o limbă, în scris sau vorbită, să detecteze formele particulare dintr-o imagine ori să caracterizeze spațiul tridimensional. Totodată sistemul de afișare va fi astfel realizat încît să poată determina automat modul de prezentare (textual sau grafic) pe un ecran mult mai eficient, axat pe necesitățile utilizatorului.

Multe întrebări se referă la posibilitatea ca această mașină a unui viitor nu prea îndepărtat să raționeze. Vom răspunde direct, afirmînd că mai mult ca sigur, nu în modul în care majoritatea dintre noi înțelege acest termen. Astăzi știm că pot fi construite sisteme care acceptă expresii în diferite limbi ce sînt prelucrate în timp real. De asemenea cunoaștem că există posibilitatea ca sistemele care pot „vedea" și „înțelege" mediul înconjurător să fie folosite cu succes în aplicațiile cu roboți.

Sînt numai cîteva direcții în care, indiscutabil, calculatoarele din generația a cincea vor avea un cîmp larg de acțiune.

Așadar, calculatoarele din următoarea generație vor apare ca rezultat al progreselor înregistrate în microelectronică și în domeniul inteligenței artificiale, la care se adaugă realizarea și dezvoltarea sistemelor de calcul și arhitecturilor noi, cu performanțe superioare. Într-adevăr, microelectronica va determina producerea pe scară largă a unor dispozitive cu dimensiuni din ce în ce mai reduse, inteligența artificială va asigura dirijarea dezvoltării conceptelor și tehnicilor pentru sistemele inteligente, în vreme ce noile calculatoare și arhitecturi vor direcționa progresul hardware-ului, software-ului și al dispozitivelor de legătură, toate aceste elemente vizînd elaborarea sistemului complex care nu este altul decît calculatorul din generația a cincea.

dificilă și nu foarte rentabilă. Specialiștii în AI au căutat și găsit o nouă cale să conducă la realizarea calculatoarelor cu programe inteligente.

Astfel în locul unui program de uz general au fost dezvoltate mai multe programe specializate.

După 1960, cercetătorii și-au concentrat eforturile asupra tehnicilor de reprezentare, adică asupra modului de formulare a unei probleme astfel încît aceasta să fie ușor rezolvabilă, precum și asupra modului de căutare a soluțiilor în așa fel încît rezultatele să fie obținute într-un timp relativ scurt, iar programul să nu ocupe o zonă din memorie. În jurul anului 1970, cercetătorii din domeniul AI au descoperit un element destul de important și anume că puterea de rezolvare a programelor derivă din cantitatea de cunoștințe pe care o posedă acel sistem și nu din procedurile și inferențele utilizate.

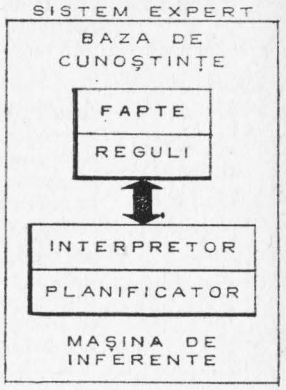



*Fig 3.1.* Schema unui sistem-expert.

Așadar, pentru soluționarea unui program inteligent este necesară dotarea acestuia cu informații specifice de înaltă calitate în strînsă legătură cu domeniul respectiv.

Denumite sisteme-expert, aceste programe au condus la apariția și, în continuare, la dezvoltarea unui nou domeniu al științei.

De ce elaborăm și dezvoltăm sisteme-expert, în loc să ne bazăm pe experții umani, așa cum am procedat în trecut? După cum rezultă din analiza elementelor prezentate în tabelul 3.1, există cîteva motive temeinice care demonstrează că utilizarea expertizei artificiale determină creșterea posibilităților pe care le oferă raționamentul uman.

Un avantaj important al expertizei artificiale se referă la caracterul său permanent.

# 3

# Sisteme-expert

## 3.1. STRUCTURA SISTEMELOR-EXPERT

Faza revoluției calculatoarelor electronice care a determinat sporirea numărului sistemelor-expert a început după 1963, odată cu realizarea calculatoarelor cu o structură hardware avansată destinată să răspundă celor mai optimiste previziuni științifice. În timp ce specialiștii în tehnologia hardware dezvoltau circuitele integrate în miniatură, cercetătorii în domeniul software-ului realizau schimbări substanțiale în structura sistemelor de operare.

Creșterea gradului de complexitate al programelor rulate pe diferitele calculatoare evoluate a pus bazele a ceea ce astăzi numim inteligența artificială. Scopul cercetărilor din domeniul AI a fost și este de a dezvolta programe care să permită calculatoarelor, într-un anumit sens, să gîndească, să dezvolte probleme în modul în care ar face acest lucru o ființă umană.

Sistemele-expert reprezintă rodul a 20 de ani de căutări pentru a defini natura programelor pentru AI. În fig. 3.1. este prezentat un model de sistem expert. În deceniul al șaselea al secolului nostru cercetătorii din domeniul AI au încercat să simuleze procesele complicate ale gîndirii prin găsirea metodelor generale de rezolvare a unei game largi de probleme. Cu toate acestea, în ciuda unor progrese interesante această strategie nu a condus la realizarea unui salt hotărîtor în domeniu. Dezvoltarea programelor de uz general a fost destul de

TABELUL 3.1. **Avantaje și dezavantaje ale expertizei artificiale comparativ cu expertiza umană**

| | Avantaje | Dezavantaje |
|---|---|---|
| Expertiza umană | creativitate<br>adaptabilitate<br>experiență senzorială<br>concentrare puternică<br>cunoștințe generale | perisabilă<br>greu de transferat<br>documentare dificilă<br>imprevizibilă<br>scumpă |
| Expertiza artificială | permanentă<br>ușor de transferat<br>documentare ușoară<br>consecvență<br>accesibilă | lipsa de inspirație<br>nu este adaptabilă<br>intrare simbolică<br>concentrare redusă<br>cunoștințe tehnice |

Expertiza umană se poate deprecia rapid, indiferent dacă este vorba despre o activitate fizică sau de natură psihică. Un expert uman trebuie să practice constant și să se perfecționeze continuu pentru a se menține fruntaș într-un anumit domeniu. Orice perioadă semnificativă de nefolosire a cunoștințelor poate afecta serios performanțele expertului. În cazul expertizei artificiale, informația, odată achiziționată, se află permanent la dispoziția utilizatorului excluzînd perioadele marcate de deteriorări accidentale ale suporturilor de informație.

Un alt avantaj al expertizei artificiale este legat de ușurința transferării sau reproducerii acesteia. Dacă transferul cunoștințelor de la o persoană la alta este laborios, necesitînd în plus un timp și experiență îndelungate, transferarea expertizei artificiale este un proces simplu de copiere. În cazul expertizei artificiale documentarea este mult mai ușoară, pe cînd în cazul expertizei umane ea este extrem de dificilă, consumă foarte mult timp și, în plus, trebuie verificată. Pentru expertiza artificială, documentarea este un proces corect de mapare între modul în care cunoștințele sînt reprezentate în sistem și descrierea limbajului folosit pentru aceste reprezentări. Comparativ cu expertiza umană, expertiza artificială permite obținerea de rezultate mai consistente și mai sigure. Ca urmare a influenței unor factori de natură emoțională, un expert uman poate lua decizii diferite în situații identice.

Un sistem-expert nu poate fi susceptibil de astfel de scăpări. În fine, comparativ cu experții umani sistemele-expert nu sînt costisitoare, mai ales în faza de utilizare. Costul lor, oarecum ridicat în etapa de dezvoltare, este compensat de costul redus în faza de operare și de simplitatea modalităților ce pot fi folosite pentru deținerea unor copii.

Din cele ilustrate mai sus rezultă multitudinea avantajelor pe care le oferă sistemele-expert în raport cu expertiza umană. Totuși, în multe aplicații, experții umani nu pot fi înlocuiți, expertiza umană fiind net superioară celei artificiale. Acest lucru nu reflectă limitarea fundamentală a inteligenței artificiale, ci starea actuală a domeniului de referință, ca de exemplu cel în care sînt din plin solicitate creativitatea, spiritul de inventivitate proprii omului. Un expert uman recunoaște informația și o poate utiliza la sintetizarea cunoștințelor noi, în timp ce sistemul-expert tinde să se comporte neinspirat, urmînd o anumită rutină. Experții umani manipulează evenimente neașteptate prin utilizarea imaginației și a unor noi perfecționări pentru rezolvarea problemelor.

Un alt domeniu în care expertiza umană excelează este procesul de învățare. Experții umani se adaptează noilor condiții, adoptînd strategiile corespunzătoare. Sistemele-expert nu pot „învăța" noile concepte și reguli, deoarece, probabil, acest proces reprezintă o sarcină deosebit de dificilă pentru AI.

Pînă în prezent s-au realizat progrese însemnate în dezvoltarea programelor destinate instruirii ; totuși aceste programe tind să lucreze deocamdată la niveluri extrem de simple și nu dau satisfacție atunci cînd sînt confruntate cu probleme mai complexe. Experții umani dețin capacitatea de a utiliza în mod direct senzorii complecși de învățare — văzul, auzul. simțul tactil și mirosul —, în timp ce sistemele-expert manipulează simboluri care reprezintă idei și concepte ; astfel este necesar ca senzațiile să fie transformate în simboluri ce pot fi înțelese de către sistem.

O parte importantă a informațiilor se pierde în procesul de translatare, în special cînd imaginile sînt formate din seturi de obiecte și de relații între acestea. Ex-

perții umani și, în general, toți oamenii posedă un bagaj de cunoștințe generale, care acoperă un spectru larg de domenii.

Datorită cantității enorme de cunoștințe generale, nu există o cale ușoară pentru introducerea acestora într-un program inteligent, cum ar fi unul specializat ca un sistem-expert. Se apreciază că centrul funcțional al sistemului-expert constă în grupul de cunoștințe structurate să permită luarea deciziilor.

Cum sînt organizate și reprezentate aceste cunoștințe ?

Prin termenul de cunoștințe oamenii de știință din domeniul AI înțeleg informațiile necesare unui program înainte ca acesta să devină inteligent. Aceste informații pot lua forma faptelor sau regulilor, care, în cazul unui sistem-expert, nu sînt întotdeauna adevărate sau false, în sensul că uneori există un grad de incertitudine asupra validității faptelor sau corectitudinii regulilor.

Cunoștințele dintr-un sistem-expert sînt organizate în cunoștințe generale, care arată modul de rezolvare a unei probleme, și cunoștințe care prezintă modul de interacțiune cu utilizatorul. Domeniul de cunoștințe din structura sistemului formează cunoștințele de bază, în timp ce cunoștințele destinate soluționării problemelor sînt cunoscute sub denumirea generică de mașină de inferențe.

Un program care are cunoștințele astfel organizate se numește sistem bazat pe cunoaștere. Din figura 3.2. reiese că sistemele-expert sînt sisteme bazate pe cunoștințe. Baza de date din sistemul expert conține date și reguli care utilizează acele informații drept fundament pentru luarea deciziilor. Mașina de inferențe conține un

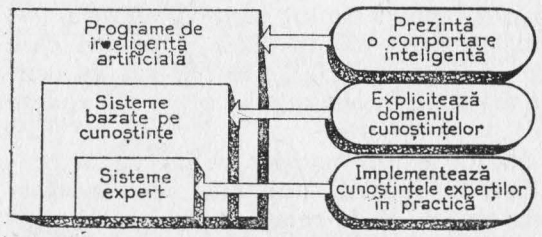



*Fig. 3.2.* Sistemele-expert — sisteme bazate pe cunoaștere.



interpretor care decide modul de aplicare a regulilor de deducție a noilor cunoștințe și un plan care stabilește ordinea în care se aplică regulile respective.

Această organizare este arătată în figura 3.3.

Modul în care sistemul utilizează cunoștințele sale este deosebit de important deoarece sistemul-expert trebuie să posede o cantitate mare de cunoștințe de înaltă calitate despre domeniul respectiv.

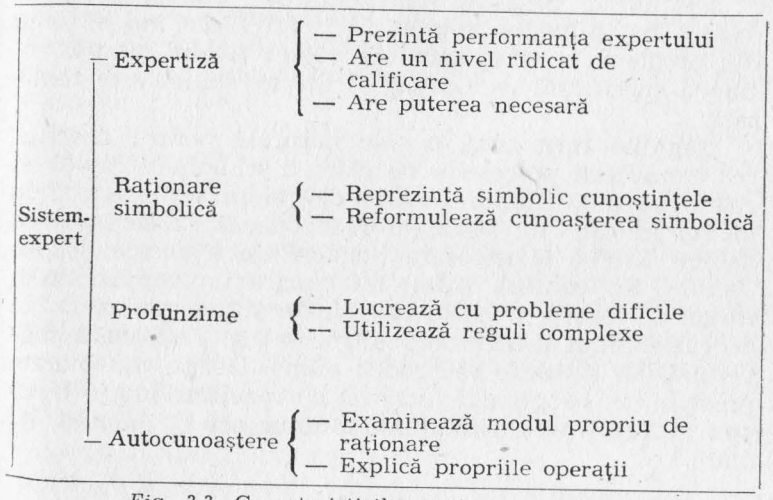



*Fig. 3.3.* Caracteristicile unui sistem-expert.

Conceptul de mașină de inferențe dă naștere la o serie de confuzii. În general, în cadrul unui sistem-expert delimitarea structurii faptelor și regulilor se realizează cu claritate ; în schimb nu este prea clară punerea la punct a mașinii de inferențe, aceasta fiind o urmare firească a dificultăților întîmpinate atunci cînd se încearcă definirea unei asemenea mașini. Modul de structurare a mașinii de inferențe depinde atît de natura domeniului de probleme, cît și de modalitățile folosite pentru reprezentarea și organizarea cunoștințelor în cadrul unui sistem-expert.

Pentru multe sisteme-expert care realizează punerea la punct a limbajelor de nivel înalt mașina de inferențe

este o parte a limbajului. Alte dispozitive bazate pe limbaje de nivel redus necesită un program special pentru proiectarea și implementarea mașinii de inferențe. Există un set standard de tehnici de reprezentare a cunoștințelor care pot fi folosite pentru realizarea unui sistem-expert. Fiecare tehnică furnizează programul cu cîteva îmbunătățiri, astfel încît acesta să fie cît mai eficient, mai ușor de înțeles și mai ușor de modificat.

Regula de bază în reprezentarea cunoștințelor se axează pe utilizarea relației IF... THEN... Cînd situația din problema curentă satisface partea IF din relație, acțiunea specificată de partea THEN a relației este realizată.

Regulile furnizează o cale naturală pentru descrierea conducerii proceselor de către o schimbare rapidă și complexă a mediului. Un set de reguli poate specifica modul în care un program reacționează la schimbarea datelor fără a fi necesare cunoștințe avansate despre fluxul de control. Într-un program convențional, fluxul de control și utilizarea datelor sînt predeterminate de către codul programului. Procesul se desfășoară secvențial, iar salturile se produc numai la anumite puncte prestabilite. Programul lucrează cu rezultate foarte bune pentru rezolvarea simultană a unui set de ecuații liniare.

## 3.2. BAZE DE DATE ȘI CUNOȘTINȚE

Sistemele-expert pot fi definite prin comparație cu programele obișnuite. Din acest punct de vedere deosebirea fundamentală dintre sistemele-expert și programele obișnuite constă în aceea că în timp ce primele lucrează cu cunoștințe, următoarele lucrează cu date.

Un sistem expert este definit ca un program caracterizat de următoarele proprietăți : expertiză, raționament simbolic, profunzime și autocunoaștere. Nivelul de expertiză atins de un asemenea sistem trebuie să fie egal cu cel atins de un expert uman. Dar simpla realizare a unor soluții valide nu este suficientă pentru caracterizarea sistemelor-expert. Comparativ cu experții



umani, aceste sisteme trebuie să ofere soluții optime într-un timp foarte scurt. Cu alte cuvinte, un sistem-expert trebuie să folosească cunoștințele de care dispune, astfel încît să găsească efectiv soluția cea mai eficientă, utilizînd diferite artificii întrebuințate de experții umani pentru eliminarea calculelor care nu sînt economice și necesare.

Pentru a imita integral un expert uman, un sistem-expert trebuie să posede nu numai profunzime într-un domeniu, ci și un orizont de cunoștințe cît mai larg. Poziția ce poate fi cîștigată prin utilizarea cunoștințelor generale și a metodelor specifice de soluționare a problemelor se poate îmbunătăți dacă raționamentul folosește principiile de bază pentru înlăturarea datelor sau regulilor incorecte.

Cînd experții umani rezolvă probleme pe care le-ar putea rezolva și un sistem-expert, aceștia nu soluționează ecuații matematice complexe, ci aleg simboluri pentru reprezentarea conceptelor problemei și aplică diferite strategii pentru manipularea acestor concepte. Un sistem-expert reprezintă, de asemenea, cunoștințele sub forma unui set de simboluri care înlocuiesc conceptele problemei.

În cadrul AI un simbol este un șir de caractere care înlocuiește un anumit concept.

Pentru a rezolva o problemă un sistem-expert manipulează mai curînd diferite simboluri, decît realizează calcule matematice standard.

Nu putem afirma că sistemele-expert nu au posibilitatea de a face calcule matematice, mai curînd trebuie subliniat faptul că acesta operează cu simboluri. În consecință, reprezentarea cunoștințelor devine un element foarte important în elaborarea sistemelor expert. Este cunoscut faptul că experții umani pot prezenta o problemă în mai multe variante, acestea fiind ulterior prelucrate astfel încît procesul respectiv să conducă la obținerea rapidă și eficientă a soluției optime. Posibilitatea de reformulare a problemei este necesară sistemelor-expert pentru a le apropia de experții umani. Din păcate, majoritatea sistemelor-expert nu posedă această

calitate. Un sistem-expert este caracterizat de profunzime, adică poate opera într-un domeniu relativ îngust, soluționînd probleme deosebit de dificile. De altfel, într-un sistem-expert regulile sînt în mod necesar complicate, fie datorită complexității lor individuale, fie datorită numărului lor ridicat. Sistemele-expert care lucrează pentru rezolvarea unor probleme ridicate de complexitatea activităților umane dețin o pondere mult mai însemnată decît sistemele expert care lucrează în domeniul denumit de specialiștii în AI al jocurilor. Dacă în primul caz datele apriori sînt asociate unei probleme cu caracter practic, fiind elaborate diverse soluții funcție de specificul activității, în al doilea caz problema de soluționat suferă o simplificare considerabilă, fără a putea fi adaptată unei anume cerințe practice.

În acest ultim context, sistemul operează cu date artificiale care, în prealabil, sînt simplificate, astfel încît problema rezultată să fie ușor de soluționat. Desigur, în această situație soluțiile obișnuite prezintă mai mult un interes din punct de vedere teoretic. Deosebirile existente între cazurile menționate sînt importante, întrucît în funcție de calea adoptată pot fi realizate sisteme- expert performante. Un sistem-expert posedă cunoștințe care îi permit să raționeze asupra propriilor operații, precum și o structură care simplifică acest proces. De exemplu, dacă un sistem-expert este organizat ca un set de reguli, atunci el poate urmări ușor lanțurile de inferențe pentru a desprinde o anumită concluzie. Dacă, în plus, sînt indicate reguli speciale care precizează modul de operare cu lanțurile de inferențe, acestea pot fi utilizate pentru verificarea calității și consistenței concluziilor. Cunoștințele de care dispune sistemul cu privire la modul în care trebuie să raționeze sînt cunoscute sub denumirea generică de metacunoștințe, ceea ce înseamnă cunoștințe despre cunoaștere.

Mai multe sisteme-expert sînt dotate cu așa-numita posibilitate de explicare. Este vorba de un ansamblu de cunoștințe destinate explicării modului în care sistemul-expert a ajuns la răspunsul respectiv. Multe din aceste explicații implică afișarea lanțurilor de inferențe și explicarea raționamentului înaintea utilizării fiecărei



reguli din lanț. Abilitatea de examinare a propriului mod de raționare reprezintă una din caracteristicile cele mai importante ale sistemelor-expert.

Autocunoașterea este importantă din următoarele motive :

— utilizatorii tind să aibă mai multă încredere atît în rezultate, cît și în sistem ;

— dezvoltarea sistemului este mai rapidă atunci cînd acesta poate fi ușor depanat ;

— presupunerile fundamentale care determină explicitarea operațiilor realizate de sistem ;

— ușurință de a prevedea și testa efectul unei schimbări în sistemul de operații.

Explicarea este numai un aspect limitat al autocunoașterii. În viitor, sistemele-expert vor fi dotate cu metacunoștințe separate și explicite, la fel cum sînt reprezentate cunoștințele propriu-zise ale sistemului-expert.

## 3.3. APLICAȚII ALE SISTEMELOR-EXPERT

Sistemele-expert au fost elaborate pentru a soluționa diferite tipuri de probleme. Principalele operații realizate de aceste sisteme sînt prezentate în tabelul 3.2. Sistemele-expert care realizează interpretarea utilizează senzori pentru a descrie o anumită situație. Aceste sisteme solicită tehnici speciale prin extragerea caracteristicilor din șiruri continue de date, forme de undă sau imagini și metode pentru reprezentarea simbolică a acestora. Sistemele destinate interpretării pot procesa tipuri diferite de date. De exemplu, atît sistemele de vedere, cît și cele de înțelegere a vorbirii utilizează intrări naturale — în primul caz vizuale, iar în al doilea semnale audio — pentru deducerea caracteristicilor și înțelesului procesului studiat. Sistemul de interpretare folosit în chimie utilizează informații despre difracția razelor X sau masa spectrală pentru deducerea structurii compușilor chimici.

TABELUL 3.2. **Categoriile de operații ale sistemelor-expert**

| CATEGORIE | MOD DE ABORDARE |
|---|---|
| Interpretare | Deducerea stării după informațiile parvenite de la senzori |
| Prevedere | Deducerea cunoștințelor corespunzătoare unor situații date |
| Diagnoză | Deducerea stării de proastă funcționare a unui sistem din observațiile făcute |
| Design | Proiectarea unor obiecte sub anumite constrîngeri |
| Planificare | Proiectarea activităților |
| Monitorizare | Compararea observațiilor cu rezultatele așteptate |
| Depanare | Prescrierea remediilor în cazul funcționării defectuoase |
| Reparație | Execută planuri de administrare a remediilor prescrise |
| Instruire | Pregătește specialiști în anumite domenii |
| Control | Guvernează comportarea sistemului |

Sistemele-expert întrebuințate în geologie interpretează măsurători despre conductivitatea rocilor în jurul puțurilor de foraj. Sistemele de interpretare utilizate în medicină folosesc măsurătorile realizate de dispozitivul de control al stării pacientului. Sistemele-expert care realizează previziunea deduc în mod corect consecințele rezultate din anumite situații date. Aceste sisteme utilizează modele de simulare și programe ce reflectă caracteristicile unei anumite activități pentru a genera, pe baza unor date particulare de intrare, diferite situații sau scenarii. Aceste situații particulare, alături de cunoștințele despre procesul care le-a produs, formează baza previziunilor.

Pînă în prezent, ca urmare a dificultăților apreciabile de realizare și interfațare cu modelele de simulare, în domeniul AI s-au dezvoltat puține sisteme-expert de previziune.



Sistemele-expert de diagnosticare utilizează descrierile eventualelor situații, caracteristicile de comportament sau cunoștințele despre proiectarea componentelor pentru a deduce cauzele probabile de funcționare defectuoasă a sistemului investigat. Exemple de astfel de sisteme sînt cele care determină cauzele probabile ale unei boli după simptomatologia prezentată de pacienți, localizează defectele în circuitele electrice și detectează componentele imperfecte din sistemul de răcire a unui reactor nuclear. Sistemele de diagnosticare sînt folosite și în scopuri de depanare. Acestea pot coopera cu utilizatorul pentru depistarea și înlăturarea unui defect.

Majoritatea sistemelor de diagnosticare au fost dezvoltate în medicină, iar în prezent se construiesc astfel de sisteme în domeniul sistemelor electronice de calcul.

Sistemele-expert destinate proiectării asociază o anumită configurație unui proces — de exemplu, înmulțirea genelor, proiectarea circuitelor integrate, crearea de molecule organice — pe baza unor restricții prealabile. Prin sinteză, sistemele-expert folosite în proiectare simulează părți componente ale proiectului de ansamblu, care ulterior sînt verificate sau testate. Acest tip de sisteme evită numeroase cercetări suplimentare datorită creării de planuri pentru produsele configurației dorite și evaluării acestora cu ajutorul restricțiilor. Sistemele menționate au numeroase aplicații în biologia moleculară și în microelectronică. Sistemele-expert destinate planificării proiectează activitățile specifice acestui domeniu ; ele stabilesc modul de desfășurare a activității respective înainte ca aceasta să înceapă. Aceste sisteme sînt mult utilizate în chimie, electronică etc.

Sistemele-expert care realizează monitorizarea compară comportarea sistemelor la un moment dat cu cea care se preconizează. Din acest punct de vedere sistemele de monitorizare observă dacă sînt confirmate așteptările referitoare la funcționarea normală sau presupunerile asupra unei posibile comportări anormale. Un factor im-

portant care influențează performanțele acestor sisteme este timpul în care sînt puse la dispoziția beneficiarului informațiile utile.

Sistemele-expert care realizează depanarea conțin tabele de corespondențe între defecte și remedierile corespunzătoare ; de obicei ele încorporează și componente de diagnosticare pentru a descoperi cauzele funcționării necorespunzătoare.

Aceste sisteme urmează un plan de administrare a unor remedii prescrise.

Pînă în prezent au fost realizate puține sisteme de depanare. Dintre cauze menționăm complexitatea operațiilor de remediere a diferitelor defecte, precum și faptul că sistemele respective necesită, în paralel cu operațiile de diagnosticare și depanare a defectelor, planificarea posibilităților de reparare funcție de anumite cerințe exprimate în prealabil.

Sistemele-expert pentru instruire asigură, pe lîngă îndrumarea activității studenților, diagnosticarea și corectarea comportamentului acestora. Sistemele de instruire dezvoltă un model al cunoștințelor pe care le deține studentul, prezentînd și modul în care aceste cunoștințe sînt folosite pentru soluționarea diferitelor probleme.

De menționat că sistemele-expert de acest gen interacționează direct cu studentul. Dintre domeniile în care sistemele-expert sînt intens folosite menționăm : agricultura, chimia, tehnica de calcul, electronica, ingineria, geologia, management, legislația, producția, matematica, medicina, meteorologia, fizica, științele naturii, controlul proceselor, tehnologia spațială etc. Remarcăm că domeniul medicinei este cel mai bine acoperit din acest punct de vedere, cel al chimiei urmîndu-l îndeaproape. În tabelul 3.3. sînt ilustrate o bună parte din sistemele-expert folosite astăzi în unele țări ale lumii, precum și domeniile de aplicabilitate. În continuare procedăm la descrierea succintă a sistemelor-expert respective, grupate pe domeniile în care sînt folosite.



TABELUL 3.3. **Aplicațiile sistemelor-expert**

| Sistemul-expert | Domeniul de aplicație | Sistemul-expert | Domeniul de aplicație |
|---|---|---|---|
| 1 | 2 | 1 | 2 |
| ABEL | medicină | DENDRAL | chimie |
| ACE | electronică | DFT | electronică |
| ACES | militar | DIAGNOSER | medicină |
| ADEPT | militar | DIALYSIS | |
| ADVISOR | matematică | THERAPY | |
| AI/COAG | medicină | ADVISOR | medicină |
| AI/MM | medicină | DIGITALIS | |
| AI/RHEUM | medicină | ADVISOR | medicină |
| AIRID | militar | DIPMETER | |
| AIRPLAN | militar | ADVISOR | geologie |
| AMUID | militar | DRILLING | |
| ANALYST | militar | ADVISOR | geologie |
| ANGY | medicină | DRUG INTER- | |
| ANNA | medicină | ACTION CRI- | |
| ARAMIS | medicină | TIC | medicină |
| ASTA | militar | DSCAS | drept |
| ATR | militar | ECESIS | tehnologie |
| ATTENDING | medicină | | spațială |
| AUDITOR | drept | EDAAS | manage- |
| BABY | medicină | | mentul infor- |
| BATTLE | militar | | mației |
| BDS | electronică | EEG ANALYSIS | |
| BLUE BOX | medicină | SYSTEM | medicină |
| CADHELP | electronică | EL | electronică |
| CARGUIDE | managemen- | ELAS | geologie |
| | tul informației | EMERGE | medicină |
| CASNET/ | medicină | EPES | militar |
| GLAUCOMA | | EVRISKO | electronică |
| CENTAUR | medicină | EXAMINER | medicină |
| CLOT | medicină | EXPERT | |
| CODES | managemen- | NAVIGATOR | militar |
| | tul informației | FAITH | tehnologie |
| COMPASS | electronică | | spațială |
| CONGEN | chimie | FALCON | controlul |
| CONPHYDE | inginerie | | proceselor |
| CRIB | calculatoare | FG 502-TASP | electronică |
| CRITTER | electronică | FOLIO | managemen- |
| CRYSALIS | chimie | | tul informa- |
| C 13 | chimie | | ției |
| DAA | electronică | FOREST | electronică |
| DART | calculatoare | GALEN | medicină |
| DART | militar | GAMMA | fizică |
| DELTA | inginerie | GA 1 | chimie |

| 1 | 2 | 1 | 2 |
|---|---|---|---|
| GCA | managementul informației | MESSAGE TRACE ANALYZER | electronică |
| GUIDON | medicină | META-DENDRAL | chimie |
| HANNIBAL | militar | MI | medicină |
| HASP | militar | MIXER | calculatoare |
| HDDSS | medicină | MODIS | medicină |
| HEADMED | medicină | MOLGEN | chimie |
| HEART | | MUD | geologie |
| IMAGE | | MYCIN | medicină |
| INTERPRETER | medicină | NAVEX | tehnologie spațială |
| HEME | medicină | NDS | electronică |
| HT-ATTEN-DING | medicină | NEOMYCIN | medicină |
| HYDRO | geologie | NEUREX | medicină |
| IDT | calculatoare | NEUROLO-GIST-1 | medicină |
| IMACS | fabricație | NPPC | inginerie |
| IN-ATE | electronică | OCEAN SUR-VEILLANCE | militar |
| INTERNIST-I/ CADUCEUS | medicină | OCSS | chimie |
| IRIS | medicină | OCULAR HERPES | |
| IR-NLI | managementul informației | MODEL | medicină |
| ISA | calculatoare | ONCOCIN | medicină |
| ISIS | fabricație | PALLADIO | electronică |
| I&W | militar | PATHFINDER | medicină |
| JUDITH | drept | PATREC | medicină |
| KNEE CAP | tehnologie spațială | PDS | controlul proceselor |
| KNOBS | militar | PEACE | electronică |
| LSD | drept | PEC | medicină |
| LEGAL ANALYSIS SYSTEM | drept | PIP | medicină |
| LES | tehnologie spațială | PLANT/cd | agricultură |
| LITHO | geologie | PLANT/ds | agricultură |
| LRS | drept | POMME | agricultură |
| MACSYMA | matematică | PROJCON | managementul informației |
| MATHLAB 68 | matematică | PROSPECTOR | geologie |
| MDX | medicină | PTRANS | fabricație |
| MECHO | fizică | PUFF | medicină |
| MECS-AI | medicină | RABBIT | managementul informației |
| MEDICO | medicină | | |
| MED 1 | medicină | RADEX | medicină |
| MES | militar | | |



| 1 | 2 | 1 | 2 |
|---|---|---|---|
| RBMS | tehnologie spațială | SPERIL-I | inginerie |
| | | SPERIL II | inginerie |
| REACTOR | inginerie | SPEX | chimie |
| REDESIGN | electronică | STEAMER | inginerie |
| RESEDA | managementul informației | SWIRL | militar |
| | | SYN | electronică |
| | | SYNCHEM | chimie |
| RPMS | tehnologie spațială | SYNCHEM 2 | chimie |
| | | SYSTEM D | medicină |
| RTC | militar | TALIB | electronică |
| RUBRIC | militar | TATR | militar |
| RX | medicină | TAXADVISOR | drept |
| R1 | calculatoare | TAXMAN | drept |
| R1-SOAR | calculatoare | THYROID MODEL | medicină |
| SACON | inginerie | TIMM/TUNER | calculatoare |
| SADD | electronică | TQMSTUNE | chimie |
| SAL | drept | TRANSISTOR SIZING SYSTEM | electronică |
| SARA | drept | TWIRL | militar |
| SCENARIO AGENT | militar | VM | medicină |
| SECS | chimie | WHEEZE | medicină |
| SEQ | chimie | WILLARD | meteorologie |
| SIAP | militar | XCON | calculatoare |
| SOPHIE | electronică | XSEL | calculatoare |
| SPAM | militar | YES/MVS | calculatoare |
| SPE | medicină | | |

### 3.3.1. MEDICINĂ

• *ABEL* (*A*cid *B*ase and *EL*ectrolyte) este folosit pentru diagnosticarea maladiilor de natură gastro-duodenală, prin evaluarea și aprecierea corespunzătoare a simptomatologiei specifice. Sistemul utilizează un model general al maladiilor posibile, pe baza căruia formulează întrebări, dirijînd — funcție de răspunsurile obținute — procesul de diagnosticare.

• *AI/COAG* asistă medicii în diagnosticarea bolilor de natură hemostatică prin analiza și interpretarea testelor de laborator referitoare la coagularea sîngelui. De menționat că sistemul operează cu 6 tipuri de teste privind coagularea și solubilitatea cheagurilor de uree.

• *AI/RHEUM* — implementat în EXPERT — este folosit în reumatologie pentru diagnosticarea diverselor maladii ale țesuturilor de legătură.

• *ANNA* este folosit în stabilirea dozei corespunzătoare de digitală ce trebuie administrată pacienților cu boli ale aparatului cardiovascular, precum și a frecvenței de administrare a dozei prescrise. Sistemul, implementat în LISP, urmărește reacția pacientului la medicament, ajustînd doza pînă cînd se obține răspunsul dorit.

• AI/MM (*A*rtificial *I*ntelligence/*M*athematical *M*odeling system) analizează comportarea fiziologică a rinichilor și dă explicații raționale asupra funcționării acestora. Sistemul pune întrebări asupra valorii diferiților parametri (volumul apei din organism) și interpretează observațiile (cantitatea mare de apă din organism). Expertiza realizată de *AI/MM* include legi din fizică și anatomie, principii fundamentale din fiziologie, precum și cunoștințe empirice despre procesele fiziologice. Aceste cunoștințe sînt reprezentate ca reguli care cuprind un model cauzal detaliat. Fiecare regulă conține descrieri ale principiilor sale, fundamentale pentru a putea explica funcționarea sistemului. *AI/MM* este implementat în LISP.

• *ANGY* asistă specialiștii la diagnosticarea bolilor coronariene prin identificarea și izolarea vaselor coronariene în angiograme. Mai întîi sistemul procesează diagramele digitale ale vaselor coronariene pentru extragerea caracteristicilor inițiale pe zone. Aceste informații sînt apoi interpretate de către mașina de inferențe a sistemului. Expertiza medicală este reprezentată ca reguli implementate în OPS 5 și LISP.

• *ARAMIS* (*A*merican *R*heumatism *A*ssociation *M*edical *I*nformation *S*ystem) ajută medicii în evaluarea gradului de suferință la bolnavii de reumatism. Sistemul refolosește datele anterioare despre bolile reumatice ale unor pacienți și realizează analize statistice cu aceste date. Cunoștințele sistemului ARAMIS constau dintr-o colecție de metode de analiză statistică, la care se adaugă un număr de baze de date care conțin înregistrări detaliate despre pacienți. *ARAMIS* este scris în PL/X — o formă specializată a limbajului.



• *BABY* ajută personalul medical să urmărească comportamentul și evoluția noilor născuți în unitățile de îngrijire intensivă. Sistemul încearcă să găsească modelele cele mai importante legate de seturile de informații medicale și demografice referitoare la noii născuți supuși îngrijirii intensive. BABY este scris în PASCAL.

• *CENTAUR* asistă cadrele medicale în activitatea de interpretare a testelor privind funcționarea plămînilor. Sistemul utilizează măsurătorile efectuate asupra cantității de aer din plămîni și a fluxului de aer filtrat de aceste organe pentru a determina prezența unor anomalii grave în funcționarea plămînilor. Expertiza sistemului *CENTAUR* include fiziologia pulmonară și rezultatele testelor asupra unui plămîn afectat de diferite boli. Cunoștințele sînt reprezentate ca o combinație de fapte și reguli, permițînd explicarea activității sistemului. *CENTAUR* este implementat în LISP.

• *CLOT* asistă medicii în stabilirea și evaluarea anomaliilor referitoare la coagularea sîngelui. Este implementat în EMYCIN.

• *DIAGNOSER* este destinat să contribuie la depistarea și identificarea bolilor de inimă congenitale. Sistemul sprijină activitatea cercetătorilor din domeniul AI în dezvoltarea și testarea modalităților de evaluare a naturii erorilor la stabilirea diagnosticului. DIAGNOSER conține cunoștințe asupra modului de stabilire a diagnosticului, precum și cunoștințe referitoare la anatomia și fiziologia inimii, precum și la cauzele care pot duce la apariția anomaliilor cardiace congenitale. DIAGNOSER este implementat în LISP.

• *DIALYSIS THERAPY ADVISOR* ajută medicii să selecteze un regim inițial pentru pacienții care încep să se trateze prin dializă. Sistemul trebuie să cunoască sexul pacientului, greutatea, înălțimea, volumul de urină, concentrația de uree etc. și întocmește o listă a terapiilor acceptabile. Expertiza constă din reguli corespunzătoare modului de gîndire al experților medicali. Sistemul încorporează, în plus, posibilitatea de rezolvare a unor ecuații, explică modul de raționare, mecanismul de refo-

losire a bazei de date ; de asemenea sistemul — implementat în FRANZ LISP — conține și facilități de comunicare cu utilizatorul.

• *DIGITALIS ADVISOR* recomandă specialiștilor un anumit tratament cu digitală pentru pacienții bolnavi de inimă. Sistemul pune întrebări referitoare la istoricul bolii, ritmul cardiac, nivelul potasiului etc. și, pe baza răspunsurilor, recomandă terapia de urmat. După ce pacientul a luat doza inițială, programul analizează reacția bolnavului la această doză și stabilește un nou tratament.

Expertiza, în cazul tratamentului cu digitală, este reprezentată ca o ierarhie de concepte într-o rețea semantică. Sistemul este implementat în OWLI.

• *DRUG INTERACTION CRITIC* ajută medicii în administrarea unor medicamente în condițiile în care pacientul urma deja un tratament medical pentru o altă boală. Sistemul stabilește efectele adverse și interacțiunile binefăcătoare, explică de ce se petrec aceste interacțiuni și sugerează acțiunile de corectare a efectelor adverse. Baza de cunoștințe a tipurilor de medicamente este organizată ca o ierarhie de fapte care include informații referitoare la medicamente specifice și cunoștințe despre modul de interacțiune între tratamentul cu diferite medicamente. Sistemul este implementat în PROLOG.

• *EEG ANALYSIS SYSTEM* analizează electroencefalogramele (EEG) înregistrate la pacienți prin utilizarea transformatei Fourier rapide și utilizează rezultatele obținute pentru clasificarea EEG în normale sau anormale. Sistemul utilizează cunoștințele obținute de la un encefalograf performant, reprezentate sub forma unor reguli asociate cu factori de siguranță. Sistemul explică rezultatele sale prin afișarea rezultatelor cîtorva reguli. Un program de editare a regulilor, scris în limbajul C, permite realizatorilor sistemului să aducă la zi informațiile respective.

• *ATTENDING,* folosit în învățămîntul superior, pune la dispoziția studenților în medicină informațiile necesare pentru realizarea unei anestezii într-o situație concretă dată, prin analiza critică a hotărîrilor adoptate privind administrarea anesteticelor. Prin intermediul unui pacient ipotetic modelat în prealabil, sistemul analizează planul de dozare gîndit de student, decizia sa în ceea ce privește corectitudinea sau incorectitudinea efectuării anesteziei fiind sub forma unui scurt comentariu de patru, cinci fraze.



• *BLUE BOX* — ajută medicii să selecteze cea mai bună metodă de tratament pentru pacienții suferind de depresiune psihică. Sistemul folosește cunoștințe despre simptomele acuzate de bolnavi, date despre medicația prescrisă, precum și elemente rezultate din relatările membrilor familiei bolnavului referitoare la comportamentul acestuia.

Pe această bază sistemul — implementat în EMYCIN — determină tipul și gradul depresiunii, sugerind un plan de control al acesteia. Acest plan include decizii privind spitalizarea și tratamentul medicamentos.

• *CASNET/GLAUCOMA* (*C*ausal *AS*ociation *NET* Work *GLAUCOMA*) diagnostichează asupra stadiului glaucomului și prescrie medicația corespunzătoare. Deciziile susținute de sistem referitoare la diagnostic și tratamentul indicat se bazează pe informațiile pe care acesta le deține privind simptomele prezentate de pacient, rezultatele diferitelor teste efectuate asupra bolnavului etc.

Sistemul — implementat în limbaj FORTRAN — poate descrie și interpreta cazurile ce-i sînt încredințate spre soluționare, iar în sprijinul concluziilor pe care le pune la dispoziția specialistului indică diverse lucrări de referință.

• *GUIDON* instruiește studenții în privința mijloacelor și metodelor de tratament folosite pentru pacienții spitalizați cu infecții de natură microbiană. După ce soluționează cazul încredințat, sistemul îl prezintă studentului spre rezolvare. În continuare sistemul analizează răspunsul formulat de student, îl compară cu decizia sa, apreciind corectitudinea răspunsului și, pe această bază, cunoștințele de care dispune studentul, precum și modul în care cel examinat a raționat pentru a pune corect diagnosticul și a indica medicația corespunzătoare.

• *INTERNIST—I/CADUCEOUS* asistă specialiștii din domeniul medicinei interne în diagnosticarea unor ma-

ladii, avînd ca informații primare datele privind starea generală a bolnavului, simptomatologia prezentată, precum și rezultatele analizelor de laborator. *INTERNIST —I/CADUCEOUS* este unul dintre cele mai dezvoltate sisteme-expert din domeniul medical. Sistemul — implementat în LISP — poate descrie peste 500 de boli ce se manifestă prin mai mult de 3500 forme.

• *MYCIN* ajută personalul medical în stabilirea unor terapii cît mai corecte împotriva infecțiilor microbiene. Sistemul diagnostichează cauza infecției utilizînd cunoștințe în legătură cu organismul infectat (simptome, rezultatele testelor etc.).

Sistemul — implementat în LISP — recomandă tratamentul medicamentos în concordanță cu procedura urmată de medicii cu experiență în tratarea bolilor infecțioase.

• *ONCOCIN* asistă medicii în tratarea și urmărirea maladiilor de natură canceroasă, în situația în care bolnavii respectivi sînt supuși unor experimente bazate pe chimioterapie, denumite protocoale. Sistemul — implementat în INTERLISP — selectează mijloacele și metoda de tratament funcție de numeroase informații referitoare la diagnostic, rezultatele testelor de laborator, specificul protocoalelor etc.

• *PUFF* — implementat în EMYCIN — diagnostichează prezența și gravitatea diferitelor boli pulmonare prin interpretarea măsurătorilor obținute în laboratoare specializate în explorarea funcțională a plămînilor. Datele de interpretat includ rezultatele testelor și descrierea istoricului bolii pacientului respectiv. Sistemul își fundamentează deciziile pe cunoștințele asupra rezultatelor obținute în urma efectuării testelor corespunzătoare diferitelor anomalii funcționale ale plămînilor.

• *SPE* (*S*erum *P*rotein *E*lectrophoresis) — implementat în EXPERT — distinge diferite cauze ale condițiilor de inflamare a diferitelor organe prin interpretarea unor forme de undă parvenite de la un densitometru cu afișare.

• *VM* (*V*entilator *M*anager) diagnostichează și sugerează modalitățile de reanimare după efectuarea intervențiilor chirurgicale. Sistemul identifică condițiile de alarmă, recunoaște informațiile false, caracterizează starea



pacientului și sugerează terapia corespunzătoare. De asemenea VM — implementat în INTERLISP — interpretează măsurătorile realizate de un sistem de control, pulsul, presiunea arterială și datele furnizate de un sistem de alimentare cu oxigen.

În fig. 3.4 sînt prezentate cîteva activități realizate de sistemele-expert în domeniul medicinei.

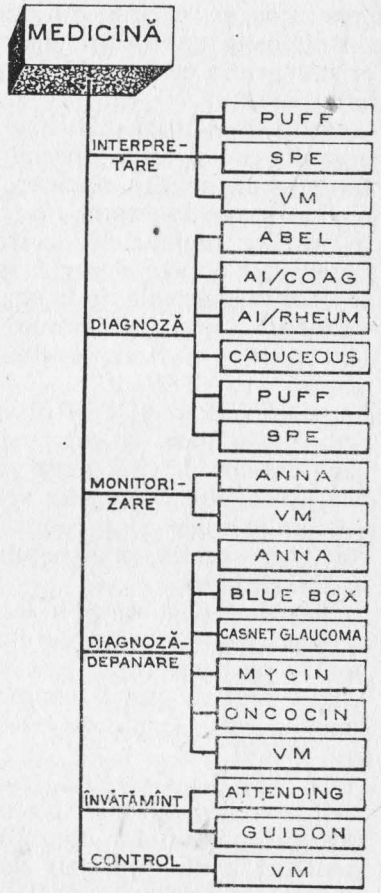



*Fig. 3.4.* Sisteme-expert din medicină.

### 3.3.2. CHIMIE

• *CONGEN* (*CON*strained *GEN*erator) este un sistem folosit pentru determinarea structurii unui compus chimic necunoscut. Datele chimice și de spectroscopie sînt furnizate de specialiști. Sistemul propune mai multe structuri posibile extrase printr-un algoritm bazat pe grafuri. CONGEN este de fapt partea de generare a ipotezelor din sistemul DENDRAL, fiind implementat ca și acesta în limbajul INTERLISP.

• *CRYSALIS* extrage structura atomică tridimensională a unei proteine din distribuția densității de electroni prin interpretarea datelor rezultate din difracția razelor X — poziția și intensitatea undelor după difracție. Pentru

generarea și testarea euristică a modelului de structură sînt folosite cunoștințe despre compoziția proteinelor și cristalografia cu raze X. Implementarea s-a făcut în limbajul LISP.

• *C-13 (Carbon-13)* determină structura compușilor organici noi și ajută specialiștii să analizeze spectrul rezonanței magnetice nucleare ale carbonului 13 prin tehnici rafinate de examinare a aranjamentului atomilor și legăturilor moleculelor complexe. Baza de cunoștințe a sistemului cuprinde reguli privind legăturile chimice și rezonanța spectrală de la structurile cunoscute. Fiind dezvoltat ca parte a sistemului DENDRAL, *C-13* este organizat într-o manieră similară. Sistemul C-13 este implementat în INTERLISP.

• *DENDRAL (DENDRitic ALgorithm)* determină structura moleculară a compușilor chimici necunoscuți, pe baza datelor privind masa spectrală și rezonanța magnetică, folosind un algoritm special.

Cunoștințele sînt reprezentate prin coduri procedurale și, respectiv, prin reguli. Sistemul Dendral este scris în INTERLISP.

• *GA 1 (Geneticist's Assistant 1)* este un sistem-expert pentru analiza structurii DNA din date asupra enzimelor, pe baza unui mecanism similar cu DENDRAL. Datele eronate pot fi compensate de utilizator în cadrul unor anumite limite de toleranță. *GA 1 este* implementat în INTERLISP.

• *META-DENDRAL* determină dependența fragmentării masei spectrometrice de caracteristicile substructurale ale moleculelor. Regulile sistemului sînt extrase din structuri tridimensionale cunoscute și spectrul de masă al acestora. *META-DENDRAL* are posibilitatea reexaminării regulilor generate pentru a înlătura soluțiile multiple sau incorecte. Implementarea sistemului s-a făcut în limbajul INTERLISP.

• *MOLGEN* este un instrument pentru realizarea experiențelor de clonare a genelor în genetica moleculară, fiind însă folosit mai degrabă ca un sistem de analiză și nu ca sistem-expert operațional. Cunoștințele necesare pentru generarea planului de lucru sînt extrase din genetică și țin seama de scopul urmărit de utilizator. Controlul și reprezentările sînt orientate spre obiecte și cadre, implementarea fiind făcută în limbajele LISP și UNITS.



• *OCSS* este un sistem-expert pentru sinteza moleculelor organice complexe, pornind de la grupurile funcționale, legături, inele și redondanță sau simetria structurii, prin transformări chimice și evaluarea rezultatului obținut. *OCSS* este implementat pe un calculator DEC PDP-1.

• *SECS* (*S*imulation and *E*valuation of *C*hemical *S*ynthesis) este utilizat pentru sinteza moleculelor organice, prin generarea elementelor constructive de bază ale structurii impuse. Planul de lucru este constituit din reacții chimice aplicabile grupurilor funcționale de atomi. *SECS* urmează un traseu invers, de la structura dorită a moleculelor pînă la generarea secvenței de sinteză. Sistemul este implementat in FORTRAN.

• *SEQ* (*SEQ*uence analysis system) este un sistem-expert dezvoltat ca parte a proiectului MOLGEN pentru analiza statistică a analogiilor structurale și simetriilor din acizii nucleici. Implementarea este realizată în LISP.

• *SPEX (S*keletal *P*lanner of *EX*periments) este utilizat pentru planificarea experiențelor complexe de laborator ; fiecare etapă a planului este corelată cu tehnicile și obiectele din baza de cunoștințe a sistemului. *SPEX* a fost gîndit pentru biologia moleculară, dar poate fi aplicat și în alte domenii. Sistemul este scris într-un limbaj bazat pe cadre ca tehnică de reprezentare, denumit UNITS.

• *SYNCHEM* este un sistem-expert pentru sinteza moleculelor organice complexe, folosind cunoștințe privind reacțiile chimice. Sistemul urmează un traseu invers — de la moleculă pînă la materialele de bază. *SYNCHEM* este implementat în limbajul PL/1. Recent a fost dezvoltată o variantă, *SYNCHEM 2*, a sistemului descris mai sus.

• *SYNCHEM 2*, ca și sistemul precedent, este utilizat în sinteza moleculelor organice complexe fără intervenția chimiștilor. Implementarea s-a făcut în limbajul PL/1.

• *TQMSTUNE* (*T*riple *Q*uadrupole *M*ass *S*pectrometer *TUNE*r) este un sistem expert pentru realizarea acordului fin al spectrometrelor de masă prin interpretarea semnalelor obținute de la aparat — raportul, lățimea și forma vîrfurilor spectrale. Cunoștințele sînt reprezentate prin cadre conform limbajului KEE, în care este implementat TQMSTUNE.

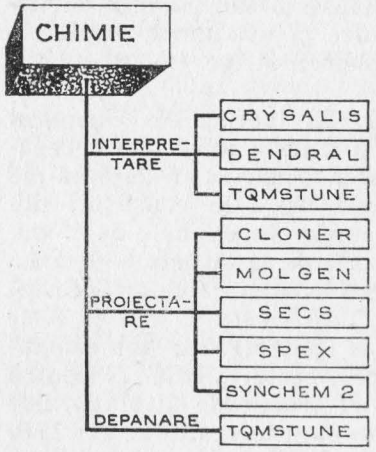



*Fig. 3.5.* Sisteme-expert din chimie.

În fig. 3.5. sînt ilustrate unele activități realizate de sistemele-expert în domeniul chimiei.

### 3.3.3. TEHNICĂ DE CALCUL

• *CRIB* (*C*omputer *R*etrieval *I*ncidence *B*ank) este un sistem-expert pentru localizarea defectelor hardware și software în tehnica de calcul. Sistemul compară descrierea utilizatorului cu informațiile din baza de date, solicitînd elemente suplimentare pînă la descoperirea modulului defect. Echipamentul analizat este modelat printr-o rețea semantică. *CRIB* este scris în limbajul CORAL 66.

• *DART* (*D*iagnostic *A*ssistance *R*eference *T*ool) este utilizat pentru diagnosticarea defectelor hardware ale calculatoarelor, pornind de la informațiile privind structura și comportarea prezumtivă a dispozitivului. Sistemul a fost aplicat în cazul circuitelor simple și al facilităților de teleprelucrare ale calculatoarelor IBM 4331. *DART* folosește o procedură de realizare a interfețelor similară cu cea a sistemului de verificare a teoremei rezoluției. Implementarea s-a făcut în limbajul MRS.

• *IDT* este un sistem-expert destinat localizării unităților defecte la calculatoare PDP 11/03. Cunoștințele sînt extrase din funcțiile componentelor unității testate



și a relațiilor stabilite între ele. Reprezentarea este realizată prin reguli, iar implementarea s-a făcut în limbajele FRANZ LISP și OPS 5.

• *ISA* (*I*ntelligent *S*cheduling *A*ssistant) planifică livrarea către beneficiari a calculatoarelor, funcție de asigurarea materialelor necesare ; se utilizează un raționament direct, într-un sistem bazat pe reguli, implementat în limbajul OPS 5.

• *MIXER* este un sistem expert destinat microprogramării pentru structurile VLSI de tipul TI 990. Sistemul generează microcoduri optimizate pentru TI 990, pornind de la descrierea microprogramului. *MIXER* folosește reprezentarea cunoștințelor ca reguli și date, implementarea fiind făcută în limbajul PROLOG.

• *R1* este o denumire a sistemului expert *XCON*, utilizat pentru stabilirea configurațiilor sistemelor de calculatoare VAX 11/780. Sistemul de reprezentare este bazat pe reguli și folosește o schemă de control cu raționament direct. *R1* este cel mai complex și solid sistem expert bazat pe reguli. Implementarea este realizată în limbajul OPS 5.

• *RL-SOAR* stabilește configurațiile sistemelor de calculatoare, avînd funcțiuni similare cu cele ale sistemului R1.

• *TIMM/TUNER* (*T*he *I*ntelligent *M*achine *M*odel *TUNER)* permite ajustarea parametrilor sistemelor de calculatoare VAX/VMS. Reprezentarea este bazată pe reguli — denumite TIMM — create în sistemul de achiziție automată a cunoștințelor.

• *XCON* (Expert Configurer of Vax 11/780. Computer Systems), cunoscut și sub denumirea de *R1*, a fost prezentat mai sus.

• *XSEL* (Expert Selling Assistant) este destinat selectării componentelor sistemelor de calculatoare electronice VAX 11/780, precum și proiectării layout-ului pentru instalarea la utilizatori. *XSEL* alege unitatea centrală, memoria internă, software-ul și echipamentele periferice, apoi intervine *XCON* pentru realizarea configurației. Sistemul este interactiv, fiind bazat pe reguli și folosind o schemă de control cu raționament direct. Implementarea s-a făcut în limbajul OPS 5.

• *YES/MVS* (*Y*orktown *E*xpert *S*ystem for *M*ultiple *V*irtual *S*torage operators) este utilizat pentru monitorizarea și controlul unuia dintre cele mai utilizate sisteme de operare, *VMS*. Sistemul funcționează în timp real, interpretează direct mesaje la MVS și transmite comenzi pentru operarea sau recomandări pentru operatorul de la consolă. Reprezentarea este bazată pe reguli cu o schemă de control cu raționament direct iar implementarea este realizată într-o variantă extinsă a limbajului OPS 5.

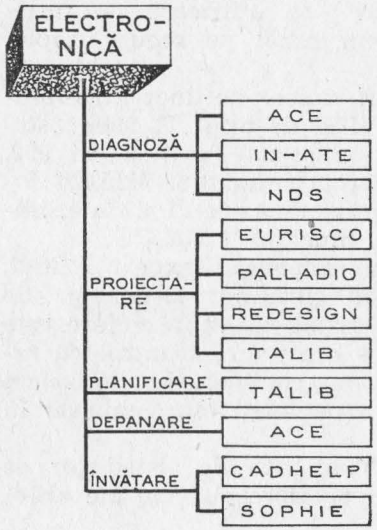



*Fig. 3.6.* Sisteme-expert din electronică.

În fig. 3.6 sînt prezentate cîteva aplicații ale sistemelor-expert în domeniul tehnicii de calcul.

### 3.3.4. ELECTRONICĂ

• *ACE* (*A*utomated *C*able *E*xpertise) permite identificarea defecțiunilor în rețelele telefonice, precum și repararea acestora fără intervenția operatorului. Reprezentarea cunoștințelor este bazată pe reguli, folosindu-se raționamentul direct. Implementarea este făcută în limbajele OPS 4 și FRANZ LISP pentru microcalculatoare 3B-2.

• *BDS* (*B*aseband *D*istribution *S*ubsystem) localizează modulele defecte în rețele de comutare a semnalelor, pornind de la strategia expertului uman și de la cunoștințele asupra structurii, funcției și relațiilor existente între diversele componente. Sistemul este implementat în limbajul LES, folosind reprezentarea prin reguli și raționamentul invers.

• *CADHELP* (*C*omputer *A*ided *D*esign *HELP*) simulează demonstrarea de către experți a modului de operare și caracteristicilor grafice pentru proiectarea asistată de calculatorul (CAD) a circuitelor digitale logice. Cunoștințele sistemului sînt organizate într-un set de subsisteme sau expertize controlate de un program de nivel înalt. *CADHELP* este scris în limbajul FRANZ LISP pentru calculatoare DEC VAX 11/780 în sistemul de operare UNIX.



• *COMPASS* (*C*entral *O*ffice *M*aintenance *P*rintout *A*nalysis and *S*uggestion *S*ystem) este utilizat pentru analiza sistemelor telefonice. Cunoștințele și datele privind structura individuală a sistemului sînt preluate de la experții în domeniu. *COMPASS* este implementat în limbajele KEE și INTERLISP-D pentru stații de lucru XEROX 1108.

• *CRITTER* este un sistem-expert pentru analiza performanțelor circuitelor VLSI, pe baza schemelor electronice și specificațiilor de intrare/ieșire. Schema circuitului este reprezentată prin cadre, formule algebrice și predicate. Implementarea este făcută în limbajul INTERLISP.

• *DAA* (*D*esign *A*utomation *A*ssistant) permite proiectarea circuitelor VLSI, pe baza descrierii fluxului de date în astfel de sisteme. *DAA* folosește reprezentarea prin reguli și raționamentul direct, fiind scris în limbajul OPS 5.

• *DFT* (*D*esign *F*or *T*estability system) este un sistem-expert destinat proiectanților de circuite VLSI, avîndu-se în vedere posibilitatea testării automate a acestora prin generarea unui model adecvat de lucru. Reprezentarea cunoștințelor este realizată printr-un cadru logic. *DFT* este implementat în limbajul PROLOG.

• *EL* asigură analiza statică a circuitelor de tipul tranzistor-diodă-rezistor și determinarea parametrilor de funcționare, cum ar fi amplificarea în tensiune. Sistemul utilizează reprezentarea prin reguli, raționamentul direct și o bază de date indexată pentru fapte și afirmații. *EL* este implementat în limbajul ARS.

• *EURISKO* este un sistem-expert care învață prin descoperire noi elemente euristice și definiții de concepte specifice domeniului analizat. Cu ajutorul sistemului *EURISKO* s-a studiat problema inventării unor dispo-

zitive microelectronice tridimensionale prin tehnica recristalizării cu raze laser, obținîndu-se rezultate interesante. Sistemul este scris în limbajul INTERLISP pentru stații de lucru din seria XEROX 1100.

• *FG 502-TASP (FG 502* Function Generator *T*roubleshooting *A*ssistant *S*ystem *P*rototype) permite diagnosticarea defectelor generatoarelor de funcții de tipul FG 502. Sistemul utilizează reguli ad-hoc pentru stabilirea comportării echipamentului și elemente euristice din experiența tehnicienilor ; este posibilă afișarea grafică a modulelor analizate, precum și ilustrarea formelor de undă care trebuie obținute în diferite puncte de măsură. *FG 502-TASP* este un sistem bazat pe obiecte, fiind implementat în SMALLTALK-80.

• *FOREST* permite izolarea și diagnosticarea defectelor în echipamentele electronice, suplimentar față de posibilitățile sistemelor de testare automată. Cunoștințele provin de la experți în domeniu, fiind codificate prin reguli, cu factori de siguranță ca la PROSPECTOR și facilități de exploatare ca la MYCIN ; *FOREST* este scris în limbajul PROLOG.

• *IN-ATE* asigură diagnosticarea defectelor la osciloscoapele de tipul Tektronix Model 465 prin analiza simptomelor și producerea unui arbore de decizie a punctelor de testare care trebuie verificate de depanator. Sistemul aplică reguli generate atît de un expert uman, cît și de modelul intern de osciloscop care este o schemă-bloc a unității controlate. Cunoștințele sînt reprezentate prin reguli, iar implementarea este realizată în limbajul FRANZ LISP.

• *MESSAGE TRACE ANALYZER* permite depanarea sistemelor cu funcționare în timp real, cum ar fi echipamentele de telecomunicații cu sute de procesoare. Elementele euristice de depanare, precum și faptele sînt reprezentate prin reguli, fiind aplicate atît prin raționament direct, cît și invers. Sistemul este scris în PROLOG și conține facilități limitate de explicare.

• *NDS* localizează defectele în rețele de comunicații prin utilizarea strategiei de diagnosticare bazate pe cunoștințe asupra topologiei și compoziției. Componentele



avute în vedere sînt procesoarele de telecomunicații, modemurile, circuitele telefonice și terminalele calculatoarelor. *NDS* este un sistem bazat pe reguli, implementat în limbajul ARBY.

• *PALLADIO,* denumit astfel după numele arhitectului Andrea Palladio (1508—1580), este un sistem-expert pentru proiectarea circuitelor VLSI n-MOS, cu ajutorul tehnicilor grafice interactive. Sistemul este implementat în limbajul LOOPS, fiind reprezentat prin obiecte, reguli sau logică.

• *PEACE* este un sistem de proiectare asistată de calculator a circuitelor electronice. Cunoștințele includ descrierea funcțională a componentelor de bază, regulile de interconectare, transformările funcționale și topologice posibile, strategiile de analiză și sinteză, precum și elementele euristice pentru prevederea greșelilor de proiectare. *PEACE* este un sistem orientat pe logică, implementat în limbajul PROLOG.

• *REDESIGN* permite reproiectarea circuitelor digitale, folosind cunoștințe reprezentate printr-o rețea de module și legături.

• *SADD (S*emi *A*utomatic *D*igital *D*esigner) asistă proiectarea circuitelor digitale, pornind de la descrierea funcțională a acestora și realizarea unui model intern. Sistemul utilizează reprezentarea cunoștințelor prin cadre și este implementat în limbajul LISP.

• *SOPHIE (SOPH*isticated *I*nstructional *E*nvironment) este un sistem-expert pentru instruirea studenților asupra depanării circuitelor electrice. Sistemul este implementat în INTERLISP și FORTRAN.

• *SYN* (Circuit *SYN*thesis Program) este destinat sintezei circuitelor electrice. Sistemul folosește constrîngeri prin manipularea algebrică a simbolurilor privind formulele care descriu componentele circuitului.

• *TALIB* realizează sinteza automată a layout-ului celulelor cu dispozitive de tipul n-MOS. Sistemul utilizează cunoștințe asupra caracteristicilor de interconectare a elementelor de circuit. *TALIB* este un sistem implementat în OPS 5, reprezentat prin reguli cu raționament direct.

• *TRANSISTOR SIZING SYSTEM* permite proiectarea circuitelor integrate în tehnologie n-MOS. Sistemul determină dimensiunile fizice ale tranzistoarelor în funcție de viteza și consumul de putere, folosind cunoștințe asupra relației viteză-putere. Reprezentarea cunoștințelor se face prin cadre și obiecte. Implementarea este realizată în limbajul LOOPS, în sistemul *PALLADIO*. În fig. 3.7 sînt prezentate cîteva aplicații ale sistemelor-expert în domeniul electronicii.

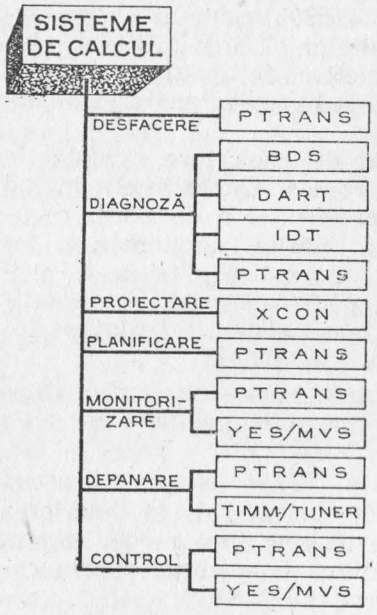



*Fig. 3.7.* Sisteme-expert din tehnica de calcul.

### 3.3.5. INGINERIE

• *CONPHYDE* (*CON*sultant for *PHY*sical Property *DE*cision) permite inginerilor chimiști să selecteze proprietățile fizice ale diferitelor substanțe prin metoda estimărilor. În cadrul acestui sistem cunoștințele sînt reprezentate prin combinarea regulilor de bază cu o rețea semantică formală. Sistemul — dezvoltat la Universitatea Carnegie-Mellon — a fost implementat în limbajul KAS.

• *DELTA* (*D*iesel-*E*lectric *L*ocomotive *T*roubleshooting *A*id) ajută personalului de întreținere să identifice și să corecteze anomaliile ce pot surveni în funcționarea locomotivelor diesel electrice prin aplicarea strategiilor de diagnosticare pentru întreținerea acestora. Sistemul DELTA a fost dezvoltat de General Electric și implementat mai întîi în LISP și apoi în FORTH.



• *NPPC* (*N*uclear *P*ower *P*lant *C*onsultant) permite operatorilor din centralele nucleare să determine cauzele unor evenimente anormale prin aplicarea regulilor de conjuncție cu modelul unei centrale funcționale.

• *REACTOR* — implementat în limbaj LISP — asistă operatorii dintr-un reactor nuclear în diagnosticarea și înlăturarea accidentelor prin controlul instrumentelor de citire, alimentării cu apă, nivelului contaminării cu radiații, deviațiilor de la condițiile normale de operare etc.

• *SACON* — dezvoltat de Universitatea Stanford (S.U.A.) — utilizează baza de cunoștințe pentru determinarea abaterilor diferitelor structuri — aripi de avioane, vasele de presiune ale reactorului nuclear, carcasa motoarelor de rachetă, poduri —, ca urmare a influenței unor condiții specifice de solicitare, în scopul elaborării strategiei potrivite pentru înlăturarea acestora. Sistemul este implementat în limbaj EMYCIN.

• *SPERI 1* evaluează avariile provocate de cutremure structurilor de rezistență ale clădirilor. Folosit în construcții, sistemul — implementat în limbaj C — utilizează cunoștințe accesate prin procedura inducției inferențelor.

• *STEAMER* este folosit în învățămîntul superior, fiind destinat instruirii studenților în domeniul propulsiei navelor. Sistemul — implementat în ZETA LISP — lucrează prin simularea pe calculator a grupului de propulsie. Prin manipularea elementelor simulate — valve, robinete și pompe — este observată variația diferiților parametri ai instalațiilor.

### 3.3.6. GEOLOGIE

• *DIPMETER ADVISOR* deduce structura geologică a scoarței terestre prin interpretarea diagramelor inclinometrelor, măsurarea conductivității rocilor în vecinătatea puțurilor de foraj ; sistemul — implementat în INTERLISP — oferă informații inclusiv asupra adîncimii puțurilor de foraj.

• *DRILING ADVISOR*, folosit chiar în timpul operațiilor de foraj, diagnostichează cele mai potrivite moda-

lități de perforare și recomandă menținerea unei anumite conduite în această acțiune în funcție de structura solului.

• *ELAS (Expert Log Analisis System)* pune la dispoziția specialiștilor modalitățile prin care pot fi controlate și interpretate rezultatele furnizate de INLAN (un program interactiv pe scară largă pentru analizarea corectă a diagramelor). Acest sistem, implementat în EXPERT, asistă utilizatorul prin recomandarea metodelor de analiză, atrage atenția asupra incorectitudinii folosirii unor căi de investigare, rezumă și interpretează rezultatele obținute.

• *HYDRO* — folosit în hidrologie — asigură utilizarea eficientă a programului HSPF destinat simulării proceselor fizice prin care precipitațiile sînt distribuite în bazinele hidrografice. Sistemul, implementat în INTERLISP, oferă date pentru aplicarea programului menționat referitoare la caracteristicile bazinelor de apă. Datele respective rezultă sub forma unor parametri numerici. *HYDRO* estimează acești parametri folosind cunoștințele de care dispune despre structura geologică a terenului, vegetație etc.

• *LITHO* asistă geologii în interpretarea datelor referitoare la densitatea rocilor, rezistivitate, propagarea sunetelor și radioactivitate. Sistemul — implementat în EMYCIN — utilizează în plus cunoștințele referitoare la regiunea geologică prospectată pentru caracterizarea rocilor aferente straturilor petrolifere din punctul de vedere al corozivității, permeabilității, compoziției, texturii, stratificării etc.

• *MUD* permite menținerea proprietăților optime ale fluidului de forare, prin diagnosticarea cauzelor ce determină alterarea lichidului de răcire — contaminarea, temperatura înaltă, presiunea ridicată, utilizarea de additive chimice necorespunzătoare — și sugerează remediile necesare. Sistemul a fost dezvoltat de Universitatea Carnegie-Mellon și implementat în OPS.

• *PROSPECTOR* ajută cercetătorii geologi în depistarea zăcămintelor de minereu. Deținînd informații despre o anumită regiune geologică, sistemul — implementat în INTERLISP — estimează probabilitatea de detectare a unor minereuri funcție de particularitățile acestora.



### 3.3.7. MANAGEMENT

• *CARGuide (Computer for Automobile Route Guidance)* servește conducătorilor auto pentru orientarea în orașe prin faptul că pune la dispoziția acestora rutele optime. Odată găsită, ruta este afișată pe un display. În timpul deplasării poziția automobilului este afișată permanent. Înainte de fiecare intersecție sistemul anunță direcția de urmat însoțită de denumirea străzii pe care urmează să ruleze vehiculul.

• *CODES (COnceptual DEsign System)* — implementat în UCILISP — permite celor care doresc să dezvolte bazele de date folosind programul IDEF 1 înțelegerea modului în care este definită schema bazei de date respective.

• *GCA (Graduate Course Advisor)*, folosit pentru instruire, facilitează întocmirea planului de învățămînt în știința calculatoarelor. Sistemul colectează informații referitoare la aptitudinile și interesele studentului, după care, asemănător unui cadru universitar, sugerează un plan de pregătire pentru respectivul student. Expertiza realizată de GCA include normele departamentale și universitare privind complexitatea programelor de pregătire, precum și descrierea cursurilor și ordinea de frecventare a acestora de către studenți.

În GCA cunoștințele sînt organizate în patru subsisteme care interacționează sub controlul unui program cu rol de dispecer. Aceste subsisteme pun la dispoziția beneficiarului numărul cursurilor care ar trebui frecventate de student, cursurile posibile, cursurile cele mai eficiente, precum și planul optim de urmat. Sistemul este implementat în PROLOG, folosind programul MYCIN în calitate de mașină de inferențe.

• *IR-NLI (Information Retrieval-Natural Language Interface)* furnizează utilizatorilor lipsiți de cunoștințe în domeniu, prin intermediul unor interfețe pentru lim-

baj natural, informații care să asigure recuperarea serviciilor oferite de către o bază de date. Cunoștințele sînt codificate ca reguli care operează pe două baze de cunoștințe — una conținînd domeniul specific (DS) de cunoștințe, iar alta, cunoștințe despre vocabular (VOC). Sistemul *IR-NLI* este implementat în FRANZ LISP și a fost realizat la Universitatea din Udine (Italia).

• *PROJCON (PROJect CONsultant)* ajută conducătorul unui proiect software să diagnosticheze problemele ridicate de proiectul respectiv și să înlăture eventualele deficiențe. Sistemul — implementat în EMYCIN — construiește modelul proiectului sub forma mai multor variante, după care afișează rezultatul diagnosticării, indicînd calea de urmat pentru alegerea variantei optime funcție de datele apriori de proiectare.

• *RABBIT* permite utilizatorilor să formuleze întrebările pentru baza de date. Acest asistent inteligent al bazei de date dirijează formularea întrebării pentru utilizatorii care nu au decît o idee vagă despre ceea ce doresc sau despre ceea ce limitează cunoștințele bazei de date. *RABBIT* este implementat în SMALLTALK și operează cu o bază de date reprezentată în KL-ONE.

### 3.3.8. PRODUCȚIE

• *IMACS (Intelligent Management Assistent for Computer System name factoring)* — implementat în OPS — asistă factorii de decizie din domeniul producției de echipamente de calcul, punîndu-le la dispoziție informații referitoare la sarcinile specifice privind procesul de conducere.

• *PTRANS*, folosit de firma DEC, permite controlul producției și distribuirii sistemelor de calcul. Sistemul utilizează informațiile conținute în comanda beneficiarului, precum și cele referitoare la activitatea uzinei pentru dezvoltarea unui plan de asamblare și testare a configurației comandate, incluzînd și data realizării sistemului. PTRANS — implementat în OPS 5 — urmărește îndeplinirea sarcinilor de producție, sugerează soluții, prevede epuizarea iminentă a unor materiale sau alte cazuri, surplusurile existente.



### 3.3.9. AGRICULTURĂ

• *PLANT/cd* este un sistem-expert la nivelul unui prototip de cercetare pentru prevederea distrugerii recoltei de cereale de către dăunători, folosind cunoștințe specifice domeniului. Sistemul utilizează o combinație de reguli și un set de programe de simulare ; cunoștințele sînt reprezentate prin reguli cu un mecanism de control prin raționament invers. Implementarea s-a efectuat în limbajul ADVISE.

• *PLANT/ds* este de asemenea un prototip de cercetare pentru stabilirea diagnosticului maladiilor plantelor de soia. Formele principale de reprezentare a cunoștințelor sînt regulile de expertiză și cele rezultate din programul de inferențe inductive, automat denumit AQ 11. Implementarea este realizată în limbajul ADVISE.

• *POMME (Pest and Orchard Manage Ment Expert system)* este un sistem-expert care dă consultații asupra modului de creștere a recoltei de mere, folosind cunoștințele patologiștilor și entomologiștilor privind fungicidele, insecticidele, precum și pericolul de îngheț, brumă sau secetă. Metodele de reprezentare sînt bazate pe reguli și cadre, implementarea fiind efectuată în PROLOG pentru calculatoare VAX 11/780.

## 3.4. PERSPECTIVE DE DEZVOLTARE A SISTEMELOR-EXPERT

În prezent, în multe din țările lumii există o preocupare constantă pentru cercetarea și dezvoltarea domeniului AI și al sistemelor-expert. O prognoză în privința progresului sistemelor-expert este ilustrată în tabelul 3.4.

TABELUL 3.4. **Perspectivele sistemelor-expert**

| Domeniul | Stadiul actual | Perspectiva intermediară | Perspectiva generațiilor viitoare de calculatoare |
|---|---|---|---|
| Baza de cunoștințe | $10^4$ — $10^5$ „relații" codificate manual pentru aplicații specifice | Modificarea bazei de cunoștințe pentru a nu putea fi divizată între diverse aplicații | $10^8$ — $10^9$ relații aplicabile universal |
| Achiziția de cunoștințe | O interferență/oră, fără mijloace standard pentru reprezentarea cunoștințelor | Cinci inferențe/oră cu mijloace diferite pentru reprezentarea cunoștințelor | Zece sau mai multe inferențe pe oră, autoinstruire. |
| Metodologia ingineriei cunoașterii | Cunoștințe ad-hoc pentru reprezentările în cazul fiecărei probleme | Reprezentări standard de cunoștințe, modele standard ale sistemelor fizice | Realizări fundamentale în codificarea cunoștințelor |
| Interfața utilizator-sistem | Limbaje, stilizate, grafică limitată | Module de prelucrare a limbajului și de recunoaștere limitată a vorbirii | Limbaje complet naturale, recunoașterea vorbirii și a imaginilor de către calculator. |

Cel mai important aspect al procesului de dezvoltare a sistemelor-expert se referă la codificarea cunoștințelor. Trebuie precizat că, datorită măsurilor adoptate în diferite state, a fost creată o bază largă și sofisticată de cunoștințe, existînd posibilitatea ca aceste cunoștințe acumulate și stocate să fie distribuite la cerere celor interesați fără a exista o legătură cu un anumit sistem de calcul.

În acest sens este necesară dezvoltarea așa-numitelor decompilatoare de cunoștințe destinate să realizeze trans-



latarea bazei de cunoștințe în cărți și rapoarte. Desigur, într-un asemenea context se estimează că metacunoașterea va fi mult mai utilizată în sistemele-expert. Progresele remarcabile înregistrate în ultimii ani în domeniul hardware-ului calculatoarelor vor contribui semnificativ la extinderea continuă a folosirii sistemelor-expert.

Elaborarea cu cheltuieli acceptabile a unui hardware compact pentru sistemele-expert va permite specialiștilor să dezvolte un număr tot mai mare de sisteme dedicate.

Progresul și extinderea sistemelor-expert vor fi influențate și de succesul sistemelor actuale de mare performanță. Realizările obținute în prezent în domeniul tehnicii de calcul vor permite elaborarea sistemelor-expert integrate, încorporarea acestora într-o structură a unui complex hardware/software, rezultînd astfel un sistem inteligent.

De menționat că în cazul unui sistem-expert integrat nu este necesar să se facă distincție între sistemul-expert, limbajul în care acesta este scris și calculatorul care execută limbajul respectiv.

Dimensiunea hardware-ului și scăderea prețului tehnicii de calcul vor permite realizarea unor echipamente cu un grad de complexitate mereu mai ridicat. Sistemele-expert integrate pot manipula următoarele task-uri : monitorizarea și controlul operațiilor sistemului, de-

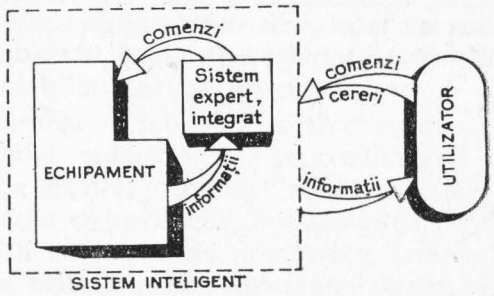



*Fig. 3.8.* Schema unui sistem inteligent.

tectarea și diagnosticarea echipamentelor defecte, asistarea în înlăturarea defectelor și organizarea lucrului în timpul cît există defectul.

În fig. 3.8. este prezentat un posibil sistem inteligent; acesta este alcătuit din echipamentul de calcul și sistemul-expert. După cum apreciază specialiștii, configurația unui sistem inteligent este corespunzătoare cînd echipamentul de monitorizare formează o ierarhie a unităților fizice aranjate într-o rețea structurală asemănătoare cu cea ilustrată în fig. 3.9.

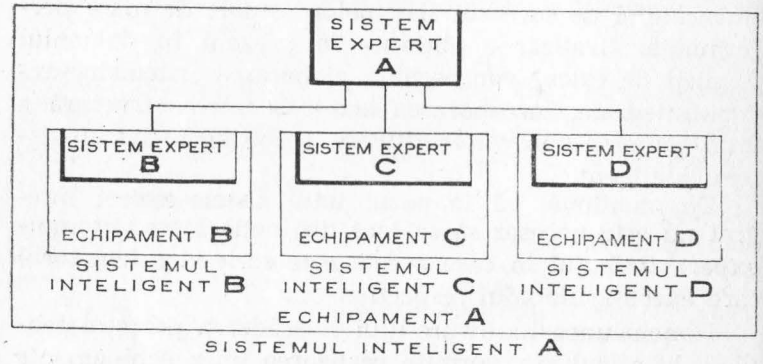



*Fig. 3.9.* Prezentarea unei ierarhii de sistem inteligent.

Fiecărei unități fizice îi poate fi atașat un sistem-expert care monitorizează modul său de operare, precum și al componentelor sale. De asemenea sistemele-expert au rolul de a decide asupra componentelor cu care urmează să lucreze în cazul unei defecțiuni.

# 4

# Roboți

Preocuparea omului de a realiza un mecanism care să imite ființele a prins contur abia în secolul nostru, odată cu proiectarea și introducerea primilor roboți. Ca și în alte domenii noi, mai ales la început, direcțiile de dezvoltare în robotică au fost multiple, de la realizarea unor dispozitive simple avînd cursa limitată, pînă la elaborarea de sisteme complexe controlate cu ajutorul calculatoarelor electronice. Roboții au fost concepuți în forme și mărimi diferite, pentru manipularea unor sarcini de cîteva grame pînă la mii de kilograme, prin acționare hidraulică, pneumatică sau electrică.

În prezent, evoluția tehnologiilor pentru roboți este strîns legată de dezvoltarea tehnicii de calcul, care a fost comparată în mod sugestiv cu progresul înregistrat într-un domeniu mai accesibil : dacă industria de automobile s-ar fi dezvoltat în același ritm, atunci un Rolls Royce fabricat acum 30 de ani ar costa acum numai două lire sterline și ar circula pe întreaga sa durată de viață numai cu jumătate de litru de benzină !

Proiectarea și lansarea în fabricație a unor roboți — în principal a roboților industriali — reprezintă și în țara noastră o direcție majoră, avîndu-se în vedere cerințele modernizării liniilor de producție, îmbunătățirii performanțelor tehnice și calitative ale produselor, reducerea consumurilor de materii prime, materiale, combus-

tibil și energie, precum și necesitatea creșterii într-un ritm mai rapid a productivității muncii.

Pentru exemplificare se prezintă robotul — manipulator pe șine cu posibilitatea programării și automatizării mișcărilor — destinat sectoarelor „fierbinți" ale Întreprinderii de Mașini Grele și „23 August" din București, „Progresul" (Brăila), Combinatul de Utilaj Greu (Cluj-Napoca) etc. Conceput de Institutul de Cercetare Științifică și Inginerie Tehnologică pentru Proiectarea Sectoarelor Calde, robotul menționat poate executa rotiri în trepte unghiulare cuprinse între 22,5° și 180° și deplasări longitudinale cu pași preselectați între 50 și 200 mm, cu o sarcină la clește între 3 și 125 tf, permițînd forjelor echipate cu asemenea sisteme să-și reducă consumul de gaz metan pentru fiecare tonă de piese forjate cu aproximativ 100 $m^3$ și să elimine aproape total efortul fizic.

De asemenea există preocupări deosebite pentru realizarea în țară a unor roboți avansați, pe baza unor tehnologii moderne și utilizarea microprocesoarelor și calculatoarelor electronice. Pentru un astfel de sistem com-

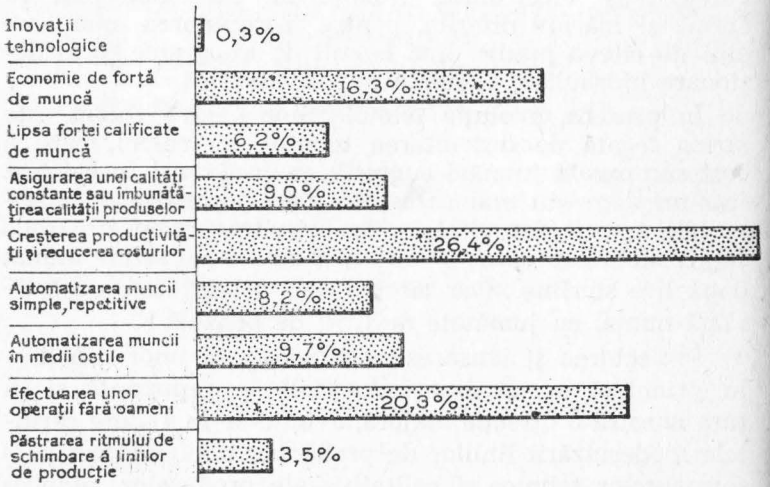



**Fig. 4.1.** Ponderea utilizării roboților în Japonia.



plex, denumit RIP 6.3, colectivul de specialiști proiectanți a primit premiul Academiei Republicii Socialiste România.

În general prin introducerea în liniile de fabricație a roboților se urmărește rezolvarea problemelor tehnico-economice ale producției pe baza celor mai bune rezultate obținute pe plan mondial. Pentru exemplificare, în fig. 4.1 sînt ilustrate principalele direcții și ponderea utilizării roboților industriali în Japonia, țara cu cel mai mare număr de roboți instalați.

În viitorul apropiat vom fi cu siguranță martorii dezvoltării accentuate a roboticii prin generalizarea folosirii microprocesoarelor, micro și minicalculatoarelor electronice, precum și a tehnicilor în curs de maturizare ale inteligenței artificiale atît pentru implementarea unor generații noi de roboți, cît și pentru conceperea unor interfețe inteligente de lucru om-mașină.

## 4.1. SCURT ISTORIC

Din cele mai vechi timpuri omul a încercat să construiască mecanisme care să imite cît mai fidel ființele. În urmă cu circa două mii de ani Hero din Alexandria a scris *Tratatul despre pneumatică*, în care a descris un mare număr de automate care se asemănau din punctul de vedere al unor însușiri cu păsările — de pildă, aceste mecanisme puteau cînta. În evul mediu automatele au fost diversificate fără a li se aduce însă modificări sau îmbunătățiri semnificative.

În acest domeniu s-au înregistrat progrese tehnice evidente abia după apariția primelor mașini mecanice de calculat, dispozitive gîndite în secolul al XVII-lea de către Pascal și, mai tîrziu, de Leibnitz ; au urmat mașinile programabile, iar în paralel au fost introduse noi facilități de programare dintre care menționăm cartelele perforate utilizate încă din anul 1805 de către francezul Jacquard în industria textilă.

Începînd din anul 1833, Charles Babbage, inventatorul mașinii *Difference*, a automatizat calculele ce se efectuau cu ajutorul unor tabele în domeniile navigației, asigurărilor și astronomiei ; el a încercat să proiecteze și o mașină „analitică" mult mai complexă cel puțin prin faptul că ar fi permis utilizarea programării prin cartele perforate. Se poate considera că începuturile roboticii moderne se regăsesc în dezvoltarea mașinilor mecanice de calculat și a războaielor de țesut.

Pe plan industrial, în anul 1873, a apărut strungul automat — prima mașină — unealtă programabilă — construit de C. M. Spencer ; în aceeași perioadă J. C. Maxwell a întreprins unul dintre cele dintîi studii despre reacția negativă care se va dovedi o noțiune fundamentală pentru controlul roboților. Robotica industrială s-a dezvoltat pornind de la astfel de idei continuate în perioadele următoare, odată cu apariția tehnicii de calcul.

Calculatorul analogic a fost realizat practic în anul 1925, calculatorul digital complet automatizat în anul 1944, iar calculatorul complet electronizat — ENIAC — în anul 1946 ; un sistem similar — *Colossus* — funcționa în mare secret în Anglia încă din anul 1944, fiind destinat descriptării mesajelor scrise în codul german „Enigma".

Cuvîntul *robot* a apărut pentru prima dată în piesa R.U.R. (Robotul Universal al lui Rossum) a cunoscutului autor ceh Karel Čapek și a început să fie utilizat pe plan internațional din anul 1923, cînd lucrarea menționată a fost tradusă în limba engleză. Creația din piesa menționată ar putea fi denumită mai exact android, dar termenul de robot s-a impus de atunci în mod definitiv în vocabularul de specialitate.

Termenul de *robotică* a fost inventat de unul dintre marii scriitori de literatură științifico-fantastică, Isaac Asimov. În anul 1942, în povestirea *Runaround*, el a utilizat pentru prima dată cuvîntul robotică și a propus celebrele ei legi :

• Un robot nu poate leza o ființă umană sau nu poate asista inactiv atunci cînd o ființă umană se află în primejdie ;



• Un robot trebuie să se supună comenzilor care i se dau de ființele umane, cu excepția acelora care ar încălca prima lege ;

• Un robot trebuie să-și protejeze propria existență, cu excepția cazurilor în care s-ar încălca prima sau a doua lege.

Cuvîntul *automatică* este mai nou decît termenul de robotică ; în anul 1950 a fost utilizat de John Diebold într-o lucrare scrisă la Harvard Business School.

De fapt, încă de la sfîrșitul anilor '40, John T. Parsons a sugerat o metodă pentru ghidarea automată a unei mașini de frezat cu ajutorul unor cartele perforate codificate pentru obținerea formelor complexe ale unei elice de elicopter. După circa 10 ani, sistemele de comandă numerică erau utilizate cu succes în producție, fiind deja începute lucrările pentru dezvoltarea unui limbaj de programare pentru folosirea lor, limbaj numit APT. Dar nici controlul numeric și nici chiar controlul numeric prin calculator — CNC (*Computer Numerical Control*) — nu aparțin roboticii deoarece nu prezintă flexibilitate față de diferite tipuri de operații, așa cum se întîmplă în cazul roboților.

Primul robot industrial a apărut în anul 1954, cînd George Devol a solicitat patent pentru un manipulator de uz general cu memorie retroactivă și control punct cu punct ; cu toate acestea robotul menționat a fost construit abia în anul 1958. Patentul a fost vîndut firmei Condec, din care s-a dezvoltat în anul 1962 firma Unimation, Inc., cea care în anul 1972 avea să devină prima firmă specializată în fabricarea roboților.

Dispozitivul de apucat și plasat a fost lansat în anul 1957 de firma Planet Corporation ; el prezenta posibilitatea programării mecanice prin roți cu came.

La începutul anilor '60 erau deja patentate principalele elemente definitorii pentru prima generație de roboți, iar în anul 1961 a fost elaborată prima instalație industrială de acest gen.

La mijlocul anilor '60 roboții puteau utiliza deja cîteva programe înregistrate, controlul fiind asigurat în mod continuu pe întreaga durată a operațiilor executate.

În aceeași perioadă s-au dezvoltat laboratoarele de cercetări în domeniul roboticii.

În consecință, la începutul anilor '70 fuseseră deja construite mai multe variante ale robotului mobil „inteligent" — denumit *Shakey* — echipat cu cameră de televiziune, detector de proximitate etc., pentru a putea să-și coordoneze mișcările în jurul obstacolelor și să plaseze diverse cutii după un model impus.

Primii roboți, deși încorporau hardware-ul calculatorului, cum ar fi memoria, foloseau numai componente electronice logice conectate prin fire în scopul realizării diferitelor operații pentru care au fost proiectați. În anul 1968 specialiștii au făcut legătura între un robot și un calculator PDP-6 de uz general, realizîndu-se astfel primul robot flexibil. Mai tîrziu, în anul 1974, a fost comercializat primul robot controlat prin minicalculator, denumit T3. Astfel de roboți, care includeau și partea de software, erau mult mai puternici decît mașinile cu circuite electronice specializate, lucrînd în cîteva sisteme de coordonate diferite cu posibilitatea de a reacționa la senzori și de a utiliza tehnici sofisticate de învățare.

În mod surprinzător, un raport prezentat în Anglia (1973), în fața Consiliului Cercetării Științifice, afirma ca o concluzie, că inteligența artificială nu va avea implicații asupra progresului roboticii. Această concluzie — desigur eronată — a dăunat cercetărilor în domeniu, contribuind la eliminarea Angliei din rîndul țărilor avansate din acest punct de vedere.

În celelalte țări realizările urmau un flux continuu. În S.U.A., de exemplu, a început să fie comercializat unul dintre primii roboți de asamblare realizat cu un braț specializat, tip PUMA. La Stanford a fost elaborată (1973) prima stație de asamblare cu un complex integrat robot-calculator. În anul 1976 a apărut încheietura flexibilă pentru ajustări fine ale brațului robotului în cursul poziționării unor subansamble ; în același timp capsula robot Viking 1, construită de NASA, ajungea pe planeta Marte.

Trecerea în faza integrării pe scară largă — LSI (*L*arge *S*cale *I*ntegration), la sfîrșitul anilor '60, a condus la creșterea capacității calculatorului electronic și, deci, la mărirea puterii de calcul. Din acești ani — perioada calculatoarelor din generația a treia — progresul roboților a fost necondiționat legat de realizările din domeniul tehnicii de calcul. Astfel, s-au dezvoltat limbajele de calculatoare pentru roboți, iar sisteme sofisticate de recunoaștere a modelelor au permis obținerea în laboratoare a detecției vizuale limitate.



La începutul anilor '80 au început lucrările pentru obținerea roboților din cea de-a doua generație, care să dețină putere de calcul suficientă pentru a face posibilă modificarea comportării lor ca răspuns la stimulii din mediul înconjurător ; devenea astfel posibilă executarea unor sarcini complexe, cum ar fi sudura cu arc sau operațiunile de asamblare cu ajutorul calculatoarelor cu circuite integrate pe scară foarte largă — VLSI (*V*ery *L*arge *S*cale *I*ntegration). În acest context se poate aprecia că creșterea imediată a interesului pentru folosirea roboților pe scară industrială a generat o nouă revoluție tehnologică.

Aplicațiile roboților s-au diversificat în cea de-a doua parte a decadei anilor '60, astfel încît, în deceniul următor, robotica industrială a devenit o secțiune a ceea ce va purta denumirea de fabricație asistată de calculator — CAM (*C*omputer *A*ided *M*anufacture). Odată cu creșterea cheltuielilor destinate realizării unor activități, precum și a necesității înlocuirii forței de muncă umane în anumite sectoare caracterizate de un anumit grad de periculozitate, în condițiile cerințelor tot mai mari ale pieții pentru produse tot mai diverse, a crescut interesul pentru robotica industrială.

Spre sfîrșitul anilor '70, ca urmare a perspectivelor privind creșterea puterii de calcul în următorii cîțiva ani, multe țări au investit sume considerabile în realizarea roboților din generația a doua în speranța că implementarea unei astfel de tehnologii va avea efecte economice favorabile.

În paralel, programele de dezvoltare a calculatoarelor electronice din generația a cincea au stimulat cercetările de inteligență artificială și progresul în domeniile conexe. Astfel, datorită unei multitudini de factori, la

începutul anilor '80, a fost dat un impuls important roboticii, în general, și roboticii industriale, în special.

În ciuda faptului că robotica își are începuturile în alte țări, în prezent Japonia este, fără îndoială, pe primul loc în ceea ce privește numărul roboților din generația întîi. Cauzele sînt desigur multiple și explicarea lor nu face obiectul acestei lucrări.

Unele statistici indicau, pentru anul 1984, circa 16 500 roboți în Japonia, 8 000 în S.U.A., 4 800 în R.F.G., 1 900 în Suedia, 1 800 în Italia, 1 700 în Marea Britanie, 1 500 în Franța și peste 6 000 în țările C.A.E.R. Este interesant de menționat totuși faptul că Suedia deține locul întîi în statisticile bazate pe evaluarea raportului dintre numărul de roboți și muncitori.

La sfîrșitul anului 1985 roboții folosiți în aplicații industriale erau în majoritate din generația întîi. Totuși, odată cu mărirea bruscă a numărului operațiilor din diferitele ramuri ale industriei bazate pe sudura cu arc și asamblare, a început să crească ponderea roboților din generația a doua. Se prevede ca pînă în anul 1990 folosirea roboților în activitățile de asamblare să depășească celelalte domenii, inclusiv pe cel al sudurii, în special în industria producătoare de automobile care, în prezent, se află pe primul loc din punctul de vedere al numărului operațiilor efectuate de roboți sau cu ajutorul acestora.

## 4.2. CLASIFICAREA ȘI STRUCTURA ROBOȚILOR

Roboții — de la cei cu mișcare repetitivă între două poziții și pînă la sistemele complexe comandate de calculatoarele electronice din liniile de fabricație integrată — se caracterizează printr-o mare diversitate. Se impune, așadar, clasificarea roboților în categorii bine definite pentru cunoașterea posibilităților de lucru, deși apar unele probleme referitoare la suprapunerea extremităților gamelor.



Pentru început este necesară delimitarea clară a roboților în raport cu mașinile automatizate. Principalul element care se ia în considerare pentru diferențiere este fiabilitatea. Astfel, de exemplu, o mașină de tăiat tablă, chiar dacă este controlată prin calculator numeric — deci este programabilă —, nu asigură un grad de flexibilitate suficient de ridicat.

În mod obișnuit un robot este un dispozitiv care, într-o anumită măsură, imită omul fie ca înfățișare, fie ca posibilități de acțiune. Una din primele definiții date robotului se referă în mod direct la acest aspect : mecanism automat care se poate substitui omului pentru a efectua unele operații, fiind capabil să-și modifice singur ciclul de execuție prin detecție fotoelectrică, creier electronic, servomotor etc.

Desigur, există și alte definiții dintre care menționăm :

- Definiția Asociației roboților industriali din Japonia — JIRA (*J*apan *I*ndustrial *R*obot *A*ssociation) : dispozitiv versatil și flexibil, care oferă funcții de deplasare similare celor ale membrelor umane sau ale cărui funcții de deplasare sînt comandate de senzori și de mijloacele proprii de recunoaștere ; acest mod de a defini robotul este destul de larg deoarece include telemanipulatoarele, protezele etc. ;
- Definiția Asociației roboților din Marea Britanie — BRA (*B*ritish *R*obot *A*ssociation), cu o arie de cuprindere ceva mai restrînsă : dispozitiv reprogramabil realizat pentru manipularea și transportul pieselor, sculelor sau altor mijloace de producție prin mișcări variabile programate pentru a îndeplini sarcini specifice de fabricație.
- Definiția Institutului de roboți american RIA (*R*obot *I*nstitute of *A*merica) similară cu precedenta : manipulator multifuncțional, reprogramabil destinat deplasării materialelor, pieselor, sculelor sau altor dispozitive specializate prin mișcări variabile programate pentru a îndeplini diferite sarcini.

Prin Norma franceză înregistrată, NF E 61-100, se definește mai întîi manipulatorul ca fiind un mecanism compus în general din elemente în serie, articulate sau

culisante unul față de altul, al cărui scop este apucarea și deplasarea obiectelor după mai multe grade de libertate ; manipulatorul este multifuncțional și poate fi comandat direct de un operator uman sau prin sisteme logice (sistem cu came, logică pneumatică, logică electrică prin cablare sau programată). Pe această bază se definește robotul ca un manipulator automat cu poziție aservită, reprogramabil, polivalent, capabil să poziționeze și să orienteze materialele, piesele, sculele și dispozitivele specializate în cursul mișcărilor variabile, programate pentru execuția unor sarcini diverse.

Pe baza definițiilor prezentate și a unor criterii diferite se pot indica mai multe modalități de clasificare a roboților ; cîteva dintre criteriile posibile după care se poate face clasificarea sînt următoarele : sistemul de control, operațiile pe care le pot executa roboții, tehnologia de realizare, generația, informațiile sau comenzile de intrare, metoda de programare, modul de acționare.

Sistemul de control al robotului diferă de la varianta simplă, caz în care constă numai dintr-o serie de limitatoare mecanice, ajustabile sau comutatoare, la calculatoarele electronice, situație în care posedă memorii programabile cu posibilitatea înregistrării unui traseu definit cu mare precizie prin coordonate ; în acest din urmă caz poate fi realizată legătura cu un alt calculator sau cu sistemul de control al unei mașini pentru sincronizarea robotului cu secvențele operației de producție.

Roboții cu secvențe limitate sînt cei mai simpli ; sistemul de control al brațului și mîinii robotului este realizat cu limitatoare și comutatoare electromecanice. În acest mod sînt definite numai capetele cursei, fără a se cunoaște traseul urmat între cele două puncte. Modul de lucru descris a condus la denumirea de roboți *pick-and-place*. Principalele aplicații ale acestor mașini sînt turnarea, încărcarea preselor, mularea plastică și automatizările speciale.

Dezavantajele principale ale roboților cu secvențe limitate constau în numărul mic de articulații posibile, de unde și limitarea numărului de mișcări, dificultatea reprogramării, timpul mare cerut pentru instalare și lipsa de control asupra traiectoriei parcurse de braț între două puncte.



La o cursă a brațului pot fi stabilite mai mult de două puncte dar memoria, de capacitate foarte mică, limitează acest număr. Cei mai avansați roboți de tipul „pick-and-place“ — de exemplu *MHU Senior* al firmei ASEA — dispun de programe pentru memorarea a cîte opt puncte pe cursă.

Programarea se face prin deplasarea brațului robotului de către operator la capetele cursei dorite pentru fiecare mișcare și prin fixarea limitatoarelor de poziție, după care se comandă începerea lucrului. Pentru a nu se efectua curse în gol, pe mașinile deservite de robot sînt instalate comutatoare sau alte dispozitive electrice care sesizează dacă piesa poate fi apucată sau dacă a fost transferată și se așteaptă plasarea altei piese.

Cea de-a doua categorie o reprezintă roboții cu secvențe variabile ; ei pot executa imediat un set nou de mișcări prin modificarea programului. În acest sens fiecare articulație trebuie să aibă posibilitatea să se oprească în oricare din punctele cursei. Există două metode pentru realizarea acestui mod de lucru : asigurarea unei cantități limitate de energie pentru a se ajunge în punctul stabilit și, respectiv, compararea poziției atinse cu poziția programată și executarea unei noi mișcări pînă cînd diferența este minimă. În primul caz se folosesc motoare pas cu pas, iar în cel de-al doilea — servomecanisme.

Inițial secvențele variabile erau constituite dintr-un număr limitat de puncte pe cursă, între acestea mișcarea făcîndu-se în mod aleator, realizîndu-se astfel roboții punct cu punct. Pe măsură ce prețul memoriilor a scăzut, numărul punctelor a crescut, ajungîndu-se la roboții cu control prin puncte multiple.

Pentru unele aplicații, cum ar fi vopsirea sau sudura cu arc, este necesar un control riguros de-a lungul întregii traiectorii printr-un număr foarte mare de puncte. Acest lucru se realizează fie prin înregistrarea punctelor la programare, fie prin calcularea fiecărei poziții prin interpolare.

Programarea roboților cu secvențe variabile, de tip punct cu punct și, respectiv, cu traiectorie continuă, se face, în cele mai multe cazuri, prin instruirea dispozitivului de către operator, după care se repetă mișcările învățate ; de aici derivă și denumirea de roboți *play back*.

Dacă se au în vedere operațiile pe care le pot executa, roboții sînt definiți funcție de aplicația pentru care sînt folosiți : sudură, asamblare, paletizare etc.

Roboții pot fi clasificați și în funcție de tehnologia folosită pentru realizarea lor. De exemplu, dacă se consideră principalele aplicații ale roboților — manipularea materialelor, presarea, mularea prin injecție, încărcarea/descărcarea mașinilor, asamblările simple, turnarea în matrițe, vopsirea, sudura cu arc, sudura prin puncte, forjarea, paletizarea, inspectarea și asamblarea complexă — atunci primele cinci dintre acestea pot fi executate de roboții realizați în tehnologie nepretențioasă, primele șase de roboți realizați în tehnologie medie, în timp ce roboții realizați cu mijloace de înaltă tehnologie execută și ultimele șapte tipuri de aplicații. Clasificarea prezentată are în atenție următoarele caracteristici : numărul de axe de mișcare, tipul și metoda programării sistemului de control, tipul mișcării de manipulare și felul acționării.

Roboții din prima categorie — de tipul *pick-and-place* — sînt dispozitivele fără servomecanisme, cu trei sau patru grade de mișcare — jos/sus, stînga/dreapta, înainte/înapoi ; ei diferă de automate prin aceea că pot să-și modifice mișcarea prin schimbarea limitatoarelor mecanice pe fiecare axă. De asemenea pot fi controlate lungimea, secvența și viteza de lucru pe o axă dată. Roboții din a doua categorie sînt utilizați pentru încărcarea/descărcarea mașinilor sau transferul materialelor, activități unde sînt necesare operații punct cu punct pentru precizie și programare în trei dimensiuni. De obicei robotul „învață" numai limita cursei mecanice ; el alege ruta cea mai rapidă și directă, singura restricție referindu-se la capacitatea de memorie a sistemului de control. Roboții din a treia categorie sînt echipați cu manipulatoare de mare flexibilitate ; ei sînt coordonați



de sisteme de control programabile avansate, mișcîndu-se cu dexteritatea unui om. Pentru unele tipuri performanțele de precizie, viteză și flexibilitate sînt ridicate, fiecare axă de mișcare fiind controlată de servomecanisme de curent continuu cu microprocesor. Modelele realizate în tehnologia de vîrf au posibilități de vedere și tactile.

O clasificare similară cu cea precedentă se bazează pe diferitele generații din care fac parte roboții. Prima generație — cea a roboților „surzi, muți și orbi" — este foarte răspîndită în prezent. A doua generație de roboți utilizează senzori pentru a obține informații în vederea modificării comportamentului în funcție de mediul înconjurător ; acești roboți sînt încă în stadiul de laborator, foarte puțini ajungînd în fabrici. Senzorii tipici sînt constituiți de sistemele de vedere și tactile. Cea de-a treia generație, a roboților, inteligenți — aflată în curs de apariție —, se caracterizează prin aceea că dispozitivele respective pot recunoaște obiectele înconjurătoare, avînd capacitatea de a acționa funcție de informațiile obținute și de baza proprie de cunoștințe ; această generație va utiliza cu siguranță tehnicile în curs de elaborare în domeniul inteligenței artificiale și al comunicațiilor de tipul om-mașină, cum ar fi recunoașterea și sinteza vorbirii, vederea artificială etc.

În Japonia roboții sînt clasificați, în principal, după informațiile sau comenzile de intrare (*J*apan *I*ndustrial *S*tandard — JIS B. 0134), ținînd seama că manipulatorul este o mașină care are posibilități similare unor membre umane și poate muta obiectele asupra cărora acționează dintr-o poziție în alta (tabelul 4.1.).

TABELUL 4.1. **Clasificarea roboților după informațiile sau comenzile de intrare**

| Tipul robotului | Definiția |
|---|---|
| 1 | 2 |
| Manipulator manual | Un manipulator (robot) care este comandat direct de om |
| Robot cu secvențe fixe | Un manipulator (robot) care poate efectua diverse operații conform informațiilor prefixate (ordinea și condițiile de operare, precum și poziția de lucru) care nu pot fi schimbate ușor |

| 1 | 2 |
|---|---|
| Robot cu secvențe variabile | Un manipulator (robot) care poate efectua diverse operații conform informațiilor prefixate (ordinea și condițiile de operare, precum și poziția de lucru) care pot fi schimbate ușor |
| Robot cu secvențe învățate *(play back)* | Un manipulator (robot) care poate repeta o operație pe baza instrucțiunilor date de om privind secvențele, condițiile, poziția și alte informații învățate prin mișcarea roboților |
| Robot cu comandă numerică | Un manipulator (robot) care poate efectua operațiile dorite pe baza unor informații asigurate prin comandă numerică — ordinea operațiilor, poziția și alte informații —, utilizînd cartele și bandă perforate sau comutatoare digitale |
| Roboți inteligenți | Un robot care poate decide asupra acțiunii pe care trebuie să o întreprindă, în funcție de informațiile obținute prin senzorii proprii și recunoaștere |

După posibilitățile de mișcare roboții sînt clasificați în sisteme avînd coordonate cilindrice, polare, carteziene sau roboți articulați.

Clasificarea acceptată în Franța (AFRI) pentru roboții industriali se prezintă în tabelul 4.2.

TABELUL 4.2. **Clasificarea roboților după AFRI, inclusiv metoda de programare**

| Tipul robotului | Definiția |
|---|---|
| 1 | 2 |
| a) Manipulatoare manuale | Dispozitive de manipulare comandate de om și avînd cel puțin 4 grade de libertate |
| — manipulatoare cu asistență musculară | — comandă prin acțiune directă |
| — telemanipulatoare | — comandă de la distanță: |
| b) Manipulatoare automate | Dispozitive de manipulare, multiax, neasistate, cu ciclu automat |



| 1 | 2 |
|---|---|
| — manipulatoare automate cu secvențe fixe | — cu reglaj mecanic |
| — manipulatoare automate cu secvențe variabile | — programate de utilizator |
| c) Robot programabil | Dispozitive de manipulare cu 5 sau mai multe axe, asistate continuu sau punct cu punct, programabile prin instruire, limbaje și simulare |
| d) Robot inteligent | Manipulator automat asistat — programabil prin instruire sau limbaje simbolice —, capabil să analizeze modificările din mediul înconjurător și să reacționeze în consecință |

Pentru înțelegerea unor termeni folosiți în mod curent prezentăm mai jos principalele comenzi utilizate în cazul roboților:

— *Comanda punct cu punct* — comanda după care mișcarea roboților este definită de un număr finit de puncte prin care dispozitivul trebuie să treacă ; poziția și viteza în fiecare punct sînt programate ; mișcările diferitelor axe sînt coordonate între ele fie simultan, fie secvențial ;

— *Comanda pe cale continuă* — comanda după care toate pozițiile și orientările, precum și viteza de-a lungul traiectoriei de parcurs sînt diferite ;

— *Comanda adaptivă* — comanda prin care parametrii acesteia sînt determinați în mod continuu și automat ca răspuns la variabilele măsurate de operația în curs pentru a asigura rezultatul într-o aplicație dată ;

— *Comanda mobilă* — comanda de deplasare în mediul înconjurător a robotului este asigurată printr-o comandă independentă de cea proprie sau

prin intermediul unor semnale de intrare/ieșire conectate la comanda proprie ;

— *Inteligența artificială* — aptitudinea de a executa funcții cum ar fi gîndirea, previziunea, rezolvarea unor probleme, efectuarea recunoașterii unor forme, percepția mediului înconjurător, recunoașterea, înțelegerea și autoprogramarea ;

— *Programarea* — introducerea în dispozitivul de memorie a elementelor necesare pentru activitatea robotului industrial ; această funcție poate fi asigurată prin programare liniară sau prin instruire.

— *Programarea liniară :*

- *analitică* — programarea liniară prin definirea matematică a traiectoriei și secvențelor ;
- *prin definirea obiectivului* — programarea liniară efectuată în funcție de sarcina de îndeplinit ; nici o traiectorie nu este importantă apriori ;

— *Programarea prin instruire* — execuția unui ciclu de lucru modelat de un operator uman ; în acest caz, operatorul poate introduce datele în memorie sub diferite forme :

- prin intermediul unui dispozitiv de comandă manuală, caracterizat prin comandă individuală a fiecărei acționări ;
- prin intermediul unui dispozitiv care permite comanda manuală intuitivă și globală a așezării și orientării extremității robotului industrial ;
- prin acțiune directă asupra extremității robotului industrial ;
- prin intermediul unei structuri avînd aceeași geometrie cu robotul industrial, pe care îl substituie.

Roboții sînt comandați de sisteme pneumatice, hidraulice sau electromecanice ; de fapt, fiecare articulație



trebuie acționată în mod separat pentru a executa mișcarea impusă.

Sistemele pneumatice sînt folosite la circa 30% dintre roboți, deși se întîlnesc, în principal, la dispozitivele cu secvențe limitate datorită prețului lor scăzut. De asemenea, în cazul comenzilor pneumatice, fiabilitatea este bună ; în plus mașinile deservite de acest gen de roboți au în general asigurat aerul comprimat. Dezavantajul major este constituit de faptul că viteza și poziția nu pot fi controlate cu precizie suficientă.

Comenzile hidraulice se obțin fie cu ajutorul sistemelor de acționare, fie cu motoare, fiind utilizate în circa 50% din aplicații. Ponderea mare a acestui sistem de comandă se datorează compactizării, fiabilității ridicate și posibilității utilizării în medii ostile, ca de exemplu în locuri unde există pericolul de explozie.

În prezent se dezvoltă dispozitivele electromecanice de comandă — servomotoarele, motoarele pas cu pas sau în impulsuri, solenoizii liniari și rotativi etc. Comenzile hidraulice și electromecanice sînt la mare concurență, dar lărgirea gamei de aplicații a condus la găsirea unor utilizări pentru fiecare din cele două tipuri, cu avantaje specifice.

Din punctul de vedere al structurii mecanice un robot este constituit din corpul propriu-zis, cunoscut și sub denumirea de trunchi sau coloană, brațul articulat, încheietura, mîna sau gripperul și, în unele cazuri, picioarele.

Corpul robotului este fixat în cele mai multe cazuri de podea, pereți sau tavanul camerei de lucru ; roboții mobili sînt montați pe diferite platforme sau — pentru anumite aplicații — chiar pe picioare.

Brațul are un rol esențial în mișcarea mîinii sau a diverselor scule folosite în procesul manipulării ori al transferului obiectelor sau pieselor. Pentru a putea orienta un obiect în toate direcțiile sînt necesare șase grade de libertate : trei de translație — dreapta/stînga, înainte/înapoi și sus/jos — și trei de rotație — pivotare, rostogolire și rotire. Se menționează că diferitele configurații ale mîinii — care se poate închide sau deschide — nu adaugă un grad de libertate suplimentar la

cele menționate mai sus. De asemenea se precizează că nu sînt necesare întotdeauna toate cele șase grade de libertate, de multe ori renunțîndu-se la cele de rotație.

Clasificarea brațelor robotului după structură conduce la șase tipuri de bază.

Roboții în coordonate cilindrice (fig. 4.2. *a)* descriu în mișcarea lor o anvelopă de tipul unui cilindru ; brațul poate fi acționat înainte/înapoi, sus/jos sau rotit stînga/drepata. Cei mai cunoscuți sînt roboții fabricați de firmele Fanuc, Prab și Seiko.

Coordonatele sferice sau polare (fig. 4.2.*b)* conduc la generarea unei sfere prin mișcarea brațului înainte/înapoi, rotire stînga/dreapta și pivotare sus/jos. Cei mai cunoscuți roboți din această familie sînt cei denumiți *Unimate,* produși de firma Unimation.

Roboții în coordonate carteziene sau rectangulare (fig. 4.2 *c*) posedă mișcări după cele trei axe, *x, y* și *z,* executînd deplasări înainte/înapoi, stînga/dreapta și sus/jos. Aplicațiile uzuale sînt în operații de asamblare, de mare precizie. Roboți de acest gen sînt realizați de firmele IBM (RS 1 care a devenit 7565), Olivetti (Sigma) și DEA (Pragma).

Există roboți care au brațul articulat, lucrînd în coordonate de revoluție (fig. 4.2 *d*). Brațul are o structură antropomorfică și posedă articulații similare umărului și, respectiv, cotului omului, fapt ce asigură două mișcări, cea de-a treia fiind dată de rotirea coloanei de bază. Anvelopa de lucru este foarte mare, de aici rezultînd principalul avantaj — economie de spațiu. Roboții tipici sînt produși de firmele Aseă, Cincinnati Milacron și Unimation (seria Puma).

La Universitatea Yamanashi din Japonia a fost proiectată, în anul 1981, o nouă structură de robot (fig. 4.2*e),* cunoscută sub numele de SCARA (*S*elective *C*ompliance *A*asembly *R*obot *A*rm). Mișcările de bază sînt două rotații în plan orizontal și o translație sus/jos, fiind comune atît coordonatelor de revoluție, cît și celor cilindrice. Principalul avantaj îl reprezintă posibilitatea de a manipula obiecte avînd greutate ridicată — pînă la 30 kg —, dispozitivul fiind folosit în operații de asam-

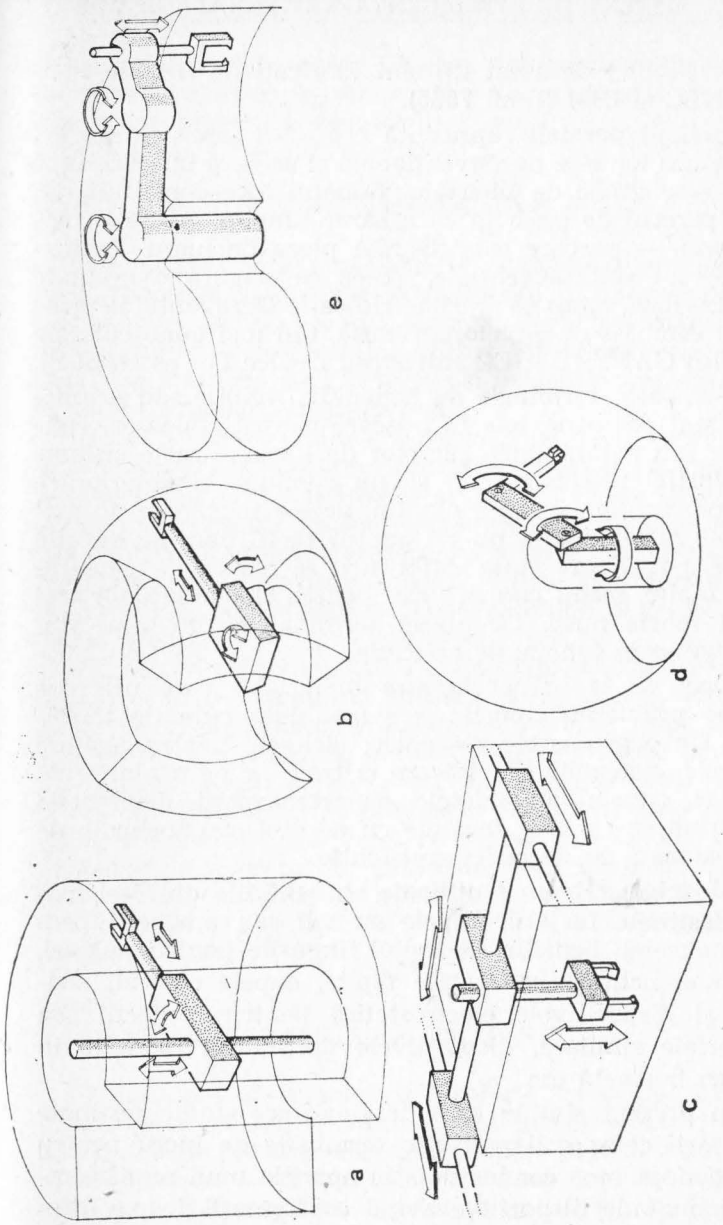



*Fig.* 4.2. Clasificarea roboților după structură (coordonate) :
*a*—cilindrice ; *b*—sferice (polare) ; *c*—carteziene (rectangulare) ; *d*—de evoluție (articulate) ; *e*—SCARA.

blare. Roboți de acest tip sînt fabricați de firmele Pentel, NEC și IBM (tipul 7535).

Roboții paraleli reprezintă cea de-a șasea categorie ; structura lor este neconvențională și asigură mișcări după cele șase grade de libertate. Robotul este constituit din trei perechi de baghete — a căror lungime poate fi modificată — pe care este dispusă placa de lucru. Articulațiile nu sînt secvențiale, ceea ce asigură o greutate mică, viteză mare și precizie ridicată. Dezavantajul principal este dat de anvelopa redusă. Cel mai cunoscut este robotul GADFLY (GEC *A*dvanced *D*evice *F*or assemb*LY*).

Sistemele terminale de acționare, mîinile sau gripperele sînt componentele în mișcare pentru apucarea, ridicarea sau manipularea pieselor de lucru ; aceste sisteme sînt astfel realizate încît să nu conducă la deteriorări. În comparație cu mîna omului, aceste terminale au mai puține articulații și nu au simțul tactil dar, pe de altă parte, pot fi proiectate astfel încît să reziste la temperaturi înalte, medii corozive sau pentru ridicarea unor greutăți foarte mari. De obicei terminalele robotului sînt construite în funcție de aplicație.

Dacă se ia în considerare materialul care urmează să fie prelucrat, tipurile cele mai cunoscute de terminale sînt ; gripperele mecanice, cîrligele pentru agățare laterală, sistemele de ridicare și transfer pe o platformă subțire, cupele sau paletele, electromagneții, elementele de prindere cu vid, degetele cu adezivi sau soclurile de tip baionetă cu deconectare rapidă.

Modelele cele mai utilizate sînt mîinile din oțel masiv destinate forjării, cupele cu vid sau magneții pentru apucarea benzilor de metal lingurile pentru pulberi, granule, lichide sau metale topite, cupele cu vid, adezivi și dispozitivele electrostatice pentru confecții sau materiale similare, dispozitivele de sudură montate în socluri baionetă etc.

În prezent sînt în curs de finalizare studii destinate realizării celor mai potrivite terminale de lucru pentru roboți, fapt ce a condus deja la apariția unui număr impresionant de dispozitive avînd configurații foarte diferite ; se asigură în acest fel cele mai bune mijloace de lucru pentru aplicații din ce în ce mai diverse în care roboții au devenit de neînlocuit.



## 4.3. INTERFAȚA OM-MAȘINĂ

Dezvoltarea fără precedent a calculatoarelor electronice, roboților și sistemelor automate a impus găsirea unor soluții pentru accesul rapid la unele informații sau la transmiterea comenzilor direct de către operatori, fără a se mai face apel la tastaturi în vederea luării unor decizii și acționării în timp real.

Recunoașterea vorbirii și sinteza vorbirii reprezintă tehnici pentru realizarea interfeței vocale om-mașină, aceasta fiind cea mai ușoară și naturală cale de lucru.

În multe cazuri au fost introduse metode noi de acces pentru informații deoarece tastaturile clasice nu mai corespundeau, prezentînd uneori dificultăți serioase, ca de exemplu pentru limba japoneză unde sînt utilizate multe semne.

Cercetările efectuate pînă în prezent au condus la satisfacerea într-o anumită măsură a cerințelor tehnologiilor de recunoaștere și sinteză a vorbirii ; pot fi recunoscute prin tehnicile actuale pînă la mii de cuvinte, iar sinteza vorbirii este mult mai precisă decît acum cîțiva ani. De asemenea progresele înregistrate de tehnologia LSI au avut ca efect reducerea apreciabilă a dimensiunilor și costurilor echipamentelor necesare, astfel încît se asigură chiar și interfețe de intrare/ieșire pentru calculatoare personale bazate pe recunoașterea și sinteza vorbirii.

### 4.3.1. RECUNOAȘTEREA VORBIRII

Avantajele principale ale vorbirii, ca mijloc de comunicație între om și mașină, sînt următoarele : timpul scurt necesar pentru instruirea operatorului, interacțiunea imediată — vorbirea este de circa două ori mai rapidă decît introducerea datelor prin tastatură —, pre-

cum și eliminarea mîinilor utilizatorului care, în acest fel, poate executa alte sarcini, uneori foarte importante, ca de exemplu în aviație.

Vorbirea se bazează pe secvențele unor segmente discrete de sunete legate în timp : aceste segmente — denumite foneme — posedă caracteristici acustice și de articulație proprii. Numărul fonemelor este limitat, deși poate fi produs un număr practic infinit de mimici de articulare. Fonemele formează unități mai mari, cum ar fi silabele și cuvintele. Cunoașterea diferențelor acustice între aceste unități ale sunetului este esențială pentru a distinge o formă de alta.

Atunci cînd sunetele vorbirii sînt legate între ele pentru a forma unități lingvistice mai mari, caracteristicile acustice ale unor foneme date se vor schimba în funcție de cele adiacente din cauza interacțiunii diferitelor structuri anatomice — limbă, buze, coarde vocale — și a gradului diferit de exprimare. Rezultatul este o suprapunere de informații, privind fonemele în semnalul acustic, de la un segment la altul.

În procesul de recunoaștere a vorbirii continue problema cea mai importantă constă în aceea că modul în care sînt pronunțate cuvintele se modifică în funcție de context, o analiză cuvînt cu cuvînt fiind practic imposibilă. În plus, în acest caz, nu sînt delimitări clare între cuvinte.

În ultimul timp s-au făcut pași importanți în fonetica articulării și fonetica acustică. Cunoașterea proprietăților acustice ale sunetelor vorbirii s-a dezvoltat de la consoane și vocale distincte la silabe, cuvinte și propoziții.

Una din metodele de bază ale prelucrării vorbirii constă în introducerea unui model standard, ca referință, în vederea comparării vorbirii curente și efectuării rapide a recunoașterii. De obicei este ales modelul cel mai apropiat de informația prezentată. Pentru clasificarea metodelor de recunoaștere a vorbirii (fig. 4.3) se iau în considerare mărimea vocabularului, legătura cu vorbitorul, condițiile de pronunțare etc.

Sistemele cu vocabular limitat — dependente de vorbitor — denumite și sisteme de recunoaștere a cuvinte-



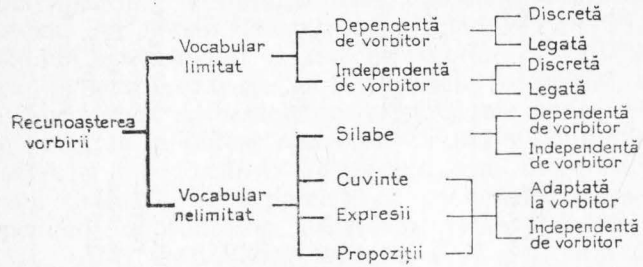



*Fig. 4.3.* Tehnicile principale ale recunoașterii vorbirii.

lor izolate, sînt cele ai căror algoritmi au fost studiați foarte bine. Principalele caracteristici ale acestor sisteme sînt date în tabelul 4.3. Cînd fiecare cuvînt este tratat ca unitate, recunoașterea impune compararea semnalelor de intrare și a celor memorate ca vocabular ; pe această bază este selectat cuvîntul care seamănă cel mai mult cu semnalul de intrare. Algoritmul metodei utilizează proceduri de aliniere în timp, dintre care cea mai cunoscută este programarea dinamică introdusă pentru prima dată în anul 1971, în Japonia.

Performanțele sistemelor ieftine, destinate recunoașterii cuvintelor izolate, diferă de la vorbitor la vorbitor din cauza metodelor relativ simple ale reprezentării și recunoașterii semnalului. Pentru recunoașterea cifrelor, de exemplu, sînt posibile precizii de peste 95%. Totuși, cînd în vocabular sînt cuvinte similare din punct de vedere acustic, precizia scade cu mult chiar pentru sisteme sofisticate sau vocabulare de dimensiuni mici.

Dezavantajele sistemelor de recunoaștere a cuvintelor izolate constau în necesitatea școlarizării operatorului și imposibilitatea reproducerii performanțelor la diferiți utilizatori ; astfel, dacă dicționarul atinge 10 000 de cuvinte procedura de școlarizare este foarte îndelungată, ceea ce face ca sistemul să devină impracticabil.

Pentru unele aplicații, cum ar fi dictarea, la care vocabularul poate fi de cîteva mii de cuvinte, sistemul trebuie să fie independent de vorbitor. Chiar și la nivelul cuvintelor izolate, recunoașterea independentă de

TABELUL 4.3. **Etape ale dezvoltării sistemelor de recunoaștere a vorbirii**

| Posibilități de recunoaștere | Cuvinte izolate dependente de vorbitor | Vorbire continuă independentă de vorbitor | Cuvinte izolate dependente de vorbitor | Cuvinte izolate independente de vorbitor | Vorbire continuă independentă de vorbitor |
|---|---|---|---|---|---|
| Sintaxa | limitată | limitată | nelimitată | nelimitată | nelimitată |
| Numărul de cuvinte ale vocabularului | 200 | 1 000 | 5 000 | 20 000 | 20 000 |
| Viteza de prelucrare cerută (Megainstrucțiuni pe secundă) | 1—10 | 100 | 300 | 1 000 | 100 000 |
| Tehnologia cerută | Model acustic și programare dinamică | Strategie de căutare de tip fascicul și algoritmi mai buni | Aproximări probabilistice și cercetări mai rapide | Constrîngeri de limbaj, reguli cantitative, semnale acustice/ foneme | Înțelegerea limbajelor naturale bază de cunoștințe și învățarea din erori |



vorbitor, dintr-un vocabular mare, constituie o problemă complexă. Un sistem dezvoltat de Universitatea Cornegie-Mellon este capabil să distingă literele din alfabetul englez. Sistemul extrage caracteristicile vocalelor, cum ar fi frecvențele și pantele formant, și ale consoanelor, ca de exemplu modul de abordare, timpul și componența spectrală, și recunoaște relații invariante între ele. În prezent, sistemul are o precizie de peste 90% în modul independent de vorbitor. Sistemul recunoaște literele cele mai ușor confundabile (b, c, d, s, g, p, t, v, și z) cu precizii de circa 90%, față de 60—70% în cazul folosirii metodei descrise anterior.

Cele mai multe sisteme de recunoaștere a cuvintelor izolate necesită ca fiecare cuvînt să fie delimitat prin pauze ; acest lucru este necesar deoarece în vorbirea continuă semnalul acustic este adesea alterat la capete, făcînd dificilă determinarea locului în care un cuvînt se termină și altul începe.

Vorbirea cu pauze între cuvinte nu este naturală, astfel că sistemele care impun acest lucru nu sînt acceptate de beneficiari. Au fost și încercări de generalizare a algoritmilor recunoașterii cuvintelor izolate pentru cuvinte legate. Metoda implică baleiajul pronunției pentru toate combinațiile posibile de cuvinte, cu condiții speciale la capetele cuvintelor.

Firma Nippon Electric Co. a dezvoltat un sistem pentru recunoașterea cuvintelor prin dubla aplicare a algoritmului.

Recunoașterea vorbirii continue este o problemă de complexitate mult mai mare decît în cazul cuvintelor izolate, deoarece proprietățile acustice ale unui cuvînt dat se pot schimba în mod semnificativ, în funcție de poziția în propoziție, astfel de schimbări fiind limitative pentru cuvinte ca pronume și prepoziții. Pe de altă parte, proprietățile acustice ale unui cuvînt pot fi modificate de cuvintele adiacente. De asemenea sintaxa, semantica și înțelegerea subiectului descris pot modifica semnalul vorbirii.

Cercetările efectuate la începutul anilor '70 au avut ca obiect recunoașterea vorbirii continue, de la mai mulți vorbitori, cu o adaptare minimă la vorbitor, care să ope-

reze cu un vocabular de 1 000 cuvinte și sintaxă artificială pentru un domeniu restrîns. Sistemul impunea o precizie semantică de peste 90% și execuția pe un calculator cu 100 milioane de instrucțiuni pe secundă. Proiectul s-a încheiat în anul 1976 cu demonstrarea mai multor sisteme, dintre care Harpy și Hearsay-II, elaborate la Universitatea Carnegie-Mellon, au depășit chiar obiectivele propuse.

Unul din cele mai avansate sisteme de recunoaștere a vorbirii este în curs de realizare la firma IBM — laboratorul Thomas J. Watson, sub conducerea lui F. Jelinek. Semnalul de intrare este digitalizat și fragmentat în segmente de ordinul sutimilor de secundă, apoi spectrul este analizat de către sistem. Fiecare segment este comparat cu un set de modele, cel mai apropiat fiind transmis pentru definirea unui cuvînt posibil. Cuvintele generate sînt testate prin crearea unei matrițe de posibilități care determină variantele cele mai bune. În paralel se introduce probabilitatea ca un cuvînt să fie într-o anumită parte a propoziției. Cele două probabilități sînt multiplicate pentru fiecare cuvînt posibil, rezultînd o ordine a tuturor posibilităților. Ciclul se repetă pentru cuvîntul următor pînă la identificarea întregii propoziții.

Rezultatele obținute de sistemul descris, pentru 5 000 cuvinte în care acestea sînt citite cu pauze între ele, reprezintă 95% pentru un număr de șase vorbitori.

Dezavantajele sistemului IBM sînt : dependența de vorbitor, care impune o durată mare de școlarizare, și viteza redusă.

În fig. 4.4. este reprezentată schema-bloc a sistemului de recunoaștere a vorbirii dezvoltat la Institutul de cercetări Matsushita ; acest sistem se bazează pe metoda de recunoaștere a cuvintelor dintr-un dicționar de mari dimensiuni, independent de vorbitor. Identificarea fonemelor și comparația cu cele 3 000 cuvinte ale dicționarului asigură o precizie de peste 95,6%.

Performanțe bune ale vorbirii, care să nu fie afectată de diferențele personale în cazul utilizării metodei cuvintelor discrete dependente, pot fi obținute prin înregistrarea modelului fiecărui cuvînt pronunțat de vorbitor. Modelul vorbirii se extinde sau se contractă în timp,



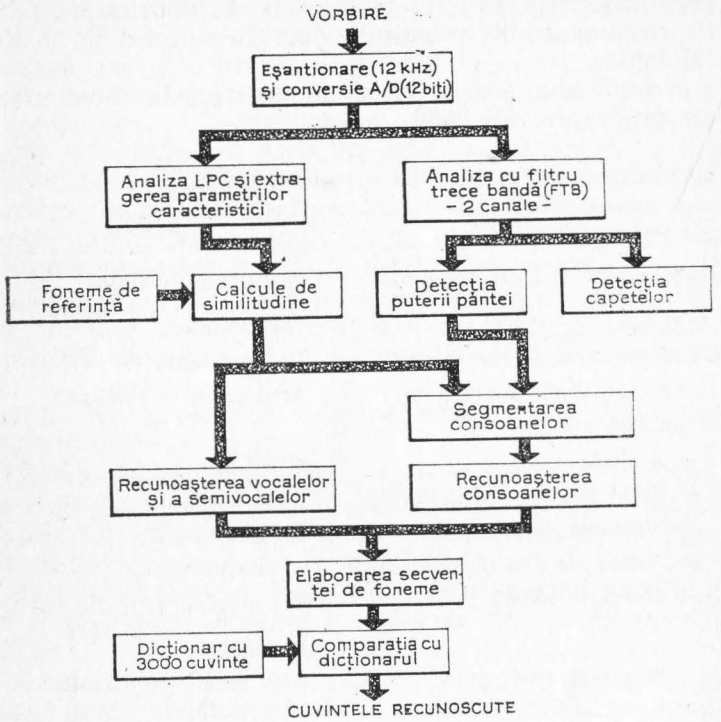



*Fig. 4.4.* Schema-bloc a sistemului de recunoaștere a vorbirii, independent de vorbitor, prin metoda fonemelor.

în funcție de viteza pronunțării. Acest dezavantaj a fost rezolvat cu ajutorul procesului de programare dinamică, dezvoltat de NEC în anul 1970, prin comercializarea circuitului LSI de tipul μPD 7764. Produsul este realizat în tehnologia NMOS, conține 40 000 de dispozitive și este capabil să compare 300 cuvinte discrete și să recunoască 30 cuvinte legate prin programare dinamică pe două nivele.

Pentru recunoașterea cuvintelor independent de vorbitor se ține seama de fluctuațiile datorate diferențelor de vorbire între persoane. Un produs tipic pentru acest mod de lucru este circuitul SR-2000 al firmei NEC destinat

recunoașterii a 16 cuvinte discrete. Echipat cu o funcție de răspuns audio, circuitul poate fi conectat la 16 linii telefonice.

În taleblul 4.4. se prezintă principalele caracteristici ale produsului SR-2000.

**TABELUL 4.4. Specificațiile circuitului SR-2000**

| Parametrul | Descrierea |
|---|---|
| Numărul de circuite telefonice | Maximum 16 linii (Telefoane cu disc, digitale sau de date) |
| Tipul recunoașterii vorbirii | Independentă de vorbitor |
| Dimensiunile vocabularului | Maximum 16 cuvinte |
| Răspunsul audio | |
| — Metoda | ADPCM |
| — Dimensiunea vocabularului | 2 048 secunde |
| — Unitatea de răspuns | 0,128 secunde |
| — Viteza de codificare a vocii | 32 kbiți/secundă |
| — Interfață locală | Semiduplex 4 800 BPS 2 linii de comunicații de date |
| — Memoria fișierului | Disc magnetic (25 MB) Discuri floppy (1 MB) |
| Opțiuni | Facsimil Ghid telecomandat Înregistrator multicanal |
| Puterea de intrare | Controlată de calculator |

Se speră să se obțină în final recunoașterea vorbirii exprimate continuu avînd un vocabular de mii de cuvinte. În acest caz cuvintele trebuie divizate în unități de bază, foneme, apoi cuvintele și propozițiile sînt recunoscute cu ajutorul fonemelor ; deci este necesară o prelucrare pe două niveluri.

Din punctul de vedere al recunoașterii vorbirii, monosilabele pot fi tratate ca și cuvintele, ceea ce a



permis comercializarea unor dispozitive de acest gen, în principal pentru limba japoneză.

Unul din obiectivele proiectului japonez de calculatoare din generația a cincea este de a realiza o mașină de scris care să accepte dictarea, cu un vocabular de 10 000 de cuvinte și capacitatea de a recunoaște vocea a sute de vorbitori. În acest scop este necesară soluționarea unor probleme importante — dimensiunea vocabularului, independența față de vorbitor, identificarea și eliminarea erorilor, viteza de lucru și costul.

Utilizarea unei asemenea mașini de scris pentru dictare impune un preț de cost foarte ridicat și, totodată, lucrul cu viteza de vorbire normală, cu reacție inversă imediată, pentru care este necesară o putere de calcul de 100 MIPS, care s-ar putea reduce prin folosirea structurilor VLSI de uz special.

Soluția la problemele mai sus menționate nu va putea fi găsită folosind tehnologiile actuale de recunoaștere a vorbirii. Optimismul existent însă în cercurile de specialiști este generat de posibilitățile tehnicilor de inteligență artificială pentru realizarea sistemelor expert și de cunoștințele care se acumulează din analiza vocii umane.

Experiențele care au avut loc recent la MIT și Carnegie-Mellon pornesc de la spectrogramele semnalelor vorbirii, care sînt identice cu cele ale semnalelor acustice.

În timp ce înțelegerea limbii vorbite nu poate fi exprimată prin reguli explicite, citirea spectrogramelor conduce la reguli cantitative, care pot fi apoi programate în sisteme de recunoaștere a vorbirii. Cu un asemenea set de reguli, eroarea este redusă sub 10% din numărul total de foneme din spectrograme, independent de vorbitor.

De asemenea folosirea unor constrîngeri constatate în alcătuirea cuvintelor în experiențele efectuate cu calculatoare electronice la MIT au arătat că o clasificare globală a sunetelor în categorii largi permite mașinilor să identifice circa o treime dintr-un vocabular de 20 000 de cuvinte ; restul de două treimi poate fi redus

la variante de cîte două cuvinte care urmează a fi diferențiate prin analize fonetice mai detaliate.

Realizările obținute arată că echipamentele care pot recunoaște cuvinte izolate din dicționare vaste — independent de vorbitor — pot fi obținute în circa 10 ani. Deși recunoașterea vorbirii continue, independentă de vorbitor este cu mult mai dificilă, sisteme capabile să efectueze dictări limitate ar putea fi proiectate în anii '90. Totuși, avînd în vedere experiența cîștigată, numărul și mărimea problemelor de rezolvat, este clar că vor fi necesare resurse substanțiale pentru obținerea unei mașini de dictat.

### 4.3.2. SINTEZA VORBIRII

Sinteza vorbirii poate fi divizată în două categorii, după cum vocabularul este limitat sau nelimitat (fig. 4.5).

Dacă vocabularul este limitat, atunci vorbirea umană este înregistrată în părți de cuvinte și propoziții, astfel încît sinteza să se poată face pe această bază. Metoda descrisă permite obținerea cu ușurință a vorbirii de înaltă calitate deoarece folosește pe larg vorbirea naturală. Există două metode de calcul : una în domeniul formei de undă, iar cealaltă în domeniul parametrilor.

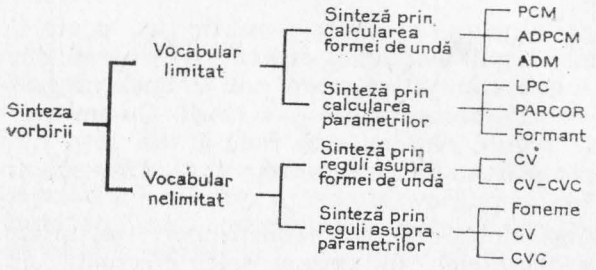



*Fig. 4.5.* Clasificarea metodelor de sinteza vorbirii.

Cu ajutorul formelor de undă se obține o vorbire de calitate excelentă deoarece sînt reproduse chiar formele de undă ale vorbirii naturale. Metodele de codificare pentru înregistrare digitală a vorbirii sînt : modulația impulsurilor în cod, PCM (Pulse Coded Modulation), și diferențială adaptivă, AD (Adaptive Differential), care utilizează compresia la jumătate a informației.



Sinteza prin calculul parametrilor are în vedere parametrii obținuți prin codificare predictivă liniară, LPC (Linear Predictive Coding), analiza formant și alte metode. Formele de undă ale vorbirii sînt sintetizate de circuitul final de sinteza vorbirii. Această metodă este foarte economicoasă deoarece reduce capacitatea de memorare necesară ; totuși, calitatea vorbirii este mai slabă decît în cazul folosirii metodei calculului formei de undă. Noile circuite LSI de sinteză au redus însă foarte mult costurile.

Sinteza vorbirii în cazul dicționarului nelimitat impune utilizarea fonemelor sau silabelor ca unități de bază. Metoda de mai sus face necesară aplicarea unei astfel de reguli de sinteză care să conducă la conectarea simplă a fonemelor sau silabelor pentru sinteza vorbirii.

Deoarece poate fi sintetizată, vorbirea conținînd orice vocabular, metoda este larg folosită și, în plus, are perspective sigure de dezvoltare ; cu toate acestea, în prezent, performanțele sintezei descrise sînt mai slabe decît în cazul calculului formei de undă sau al parametrilor.

Cele două metode prezentate — în domeniul formei de undă sau al parametrilor — au în vedere unitățile de bază ale vorbirii naturale, consoanele (c) și vocalele (v). Deși este inteligibilă la un nivel excelent, sinteza în domeniul formei de undă nu permite controlul flexibil al înălțimii sunetelor. Sinteza în domeniul parametrilor poate produce o vorbire corectă, dar nu este inteligibilă la nivelul necesar.

Tehnicile recunoașterii și sintezei vorbirii au început să fie aplicate atît în industrie cît și la domiciliu. Recunoașterea vorbirii a început să fie utilizată în industrie pentru control, verificarea datelor de intrare etc. În viitorul apropiat se prevede accesul metodelor de recunoaștere a vorbirii în domeniul controlului roboților ; în acest scop vor fi necesare metodele care folosesc vocabulare mari în vederea înglobării unor instrucțiuni complexe. De asemenea recunoașterea vorbirii cîștigă teren în pregătirea documentelor în secretariate.

La început, sinteza vorbirii a fost utilizată la jocurile electronice, dar aplicațiile s-au extins la software-ul profesional. Firma NEC a lansat calculatoarele familiale PC-6601 și PC-6601SR cu sinteza vorbirii ca funcție standard.

### 4.3.3. PRELUCRAREA LIMBAJELOR NATURALE

Primele eforturi de prelucrare a limbajelor naturale au fost legate de realizarea mașinilor de traducere, dispozitive propuse încă din anul 1946. În anii '50 au fost întreprinse cercetări importante pentru efectuarea traducerilor cu ajutorul unor mijloace sintactice — căutarea în dicționare, substituția și rearanjarea sintaxei propoziției —, dar în prezent se consideră că metodele folosite au fost greșite.

Programele de limbaje naturale utilizînd inteligența artificială, dezvoltate în anii '60, înțelegeau propozițiile prin compararea directă cu modele avînd semnificații clare. De exemplu, programul „Eliza", creat de J. Weizenbaum de la MIT (*M*assachusetts *I*nstitute of *T*echnology), a evoluat în mod impresionant.

Un alt program, SHRDLU, a fost dezvoltat în jurul anului 1970 de către T. Winograd de la același institut și conținea anumite reguli bazate pe analogii între cuvinte și fragmente de program, sintaxa propoziției și aranjarea fragmentelor de program precum și analogii între propoziții și programe complete. Înțelegerea unei propoziții era echivalentă cu realizarea și execuția unui program complet de lucru.

În aceeași perioadă a fost conceput „Lunar", un sistem de chestionat și răspuns legat la o bază de date pentru analiza chimică a rocilor aduse de pe Lună. Echipamentul conține un sistem de analiză gramaticală, sintactică, pentru construirea unui arbore de analiză — o diagramă a propoziției —, care este utilizat pentru a pune întrebări dintr-o bază de date standard. Baza de date folosită de „Lunar" este suficient de simplă, astfel încît întrebările nu sînt ambigue ; în cazul apariției unei probleme sistemul se adresează operatorului.



La mijlocul anilor '70, un număr de sisteme de baze de date erau concepute, ca și programul „Eliza", pe gramatici semantice și nu pe analize sintactice. Sistemele Ladder și Planes constituie astfel de exemple.

Primul sistem de limbaj natural comercial, denumit Robot, a fost scris de Larry Harris și a început să fie livrat din anul 1978. Sistemul este organizat ca o analiză gramaticală sintactică și furnizează soluții originale la problemele legate de vocabularul specializat sau la ambiguități. De asemenea sistemul folosește toate cuvintele dintr-o bază de date pentru a-și forma lexicul, realizează o cercetare separată pentru fiecare citire posibilă a unei întrebări și compară apoi numărul de variante pentru a afla soluția.

Roger Shank a introdus în anul 1970 programe care să se ocupe de limbaje privind acțiunile umane, bazate pe principii diferite : propozițiile sînt transformate în structuri de date, organizate în jurul unui număr fix de primitive ale așa-numitelor dependențe conceptuale care sînt abstracții ale unei clase de verbe asociate. Primitivele au ca scop atît reprezentarea parafrazelor similare, cît și organizarea inferențelor despre proporții. Inferențele sînt de asemenea asociate cu primitivele dependențelor conceptuale. Adesea o singură propoziție corespunde unei secvențe de acțiuni.

Programul SAM, scris de Richard Cullingford, examinează textul pentru a selecta documente și apoi utilizează documentele pentru a completa acțiunile stereotipe nemenționate.

La sfîrșitul anilor '70 înțelegerea naturii limbajelor s-a schimbat radical. Dacă pînă recent limba era considerată ca o relație — adevărată sau falsă — între propoziții, pe de o parte, și lucruri și evenimente, pe de altă parte, în prezent cunoștințele în acest domeniu s-au îmbogățit. Pronunțarea lingvistică este acum văzută ca un caz special de acțiune, avînd un scop bine definit, în cadrul comun al unor acțiuni nelingvistice.

Dorința de a găsi și exploata paralelisme potențiale a condus la o creștere a popularității modelelor de prelucrare paralelă a limbajelor. În lucrări recente, bazate pe inteligența artificială, s-au creat programe de limbaje

naturale în care sensurile cuvintelor sînt la concurență, formînd grupări împreună cu alte sensuri ale cuvintelor și în legătură cu contextul, pentru a se ajunge la o reprezentare corespunzătoare a înțelesului.

Principalele etape ale prelucrării limbajelor naturale, realizate sau în curs de definitivare sînt următoarele :

— sisteme capabile de interogări ale bazelor de date cu limbaje naturale limitate ;

— sisteme de laborator care realizează analiza sintactică a celor mai multe propoziții gramaticale ;

— programe care traduc circa 90% din texte dintr-o limbă în alta ;

— sisteme comerciale care critică gramatica și stilul ;

— sisteme comerciale care pot rezuma și indexa informația din documente ;

— roboți comerciali care înțeleg instrucțiuni în limbaje naturale.

Pentru a fi utile, sistemele de limbaje naturale trebuie să fie capabile să prelucreze un vocabular de dimensiuni mari și baze de cunoștințe voluminoase. În cazul în care sînt în posesia unei cantități mari de cunoștințe limbajele naturale trebuie să opereze nu numai eficient, ci să găsească orientările pentru crearea unei baze largi de cunoștințe corespunzătoare înțelegerii și pentru extinderea progresivă a acestora.

Învățarea este incrementală ; concepte noi pot fi utilizate atît pentru prelucrarea limbajului, cît și pentru participarea la învățarea unor noi concepte. Acest tip de învățare necesită un sistem care să știe foarte multe pentru a acumula ceva nou.

Lucrări recente asupra limbajelor naturale au în vedere modelarea înțelegerii metaforelor, preocuparea asupra informațiilor neclare de intrare și judecarea plauzibilității înțelesului propozițiilor. Judecarea plauzibilității este necesară pentru ca un program să înțeleagă metaforele, umorul, minciunile sau exagerările. Este un fel de sens comun și este posibil numai pentru programe cu o bază imensă de cunoștințe. O altă arie de cercetări vizează modelarea caracterelor și înțelegerea unor texte lungi — episoade și unități narative. De asemenea este



importantă munca pentru elaborarea gramaticilor cu context liber, care utilizează reguli mai simple decît acelea ale tehnicilor curente de analiză sintactică a propozițiilor în reprezentări ale cunoștințelor.

Asemenea gramatici au dezavantajul că solicită mii de reguli ; din acest motiv cercetătorii lucrează la metareguli și alte proceduri care să permită programelor să genereze regulile cerute pentru transformarea unei propoziții particulare într-o reprezentare de cunoștințe, așa cum este necesar.

Proiectul japonez al generației a cincea de calculatoare prevede pentru limbajele naturale atingerea, pînă în anul 1992, a unor performanțe cum ar fi :

— accesul interactiv la bazele de date ;

— indexarea și raționarea proprie, automată ;

— exprimarea cunoștințelor, în principal a celor de intrare, atît prin imagini și diagrame, cît și prin mai multe mijloace tradiționale.

Pentru a obține aceste realizări este necesar nu numai să se listeze posibilitățile dorite ale unui sistem, ci și să se stabilească unele criterii ale performanțelor specifice, cum ar fi nivelul interpretării greșite permise, volumul școlarizării necesare la utilizator, cantitatea permisă de verificări interactive ale înțelegerii și modul general de acoperire a domeniului.

Pentru a înțelege vorbirea, omul ia în considerare nu numai informațiile specifice, ci și contextul în care informația este discutată ; de aceea, limba vorbită poate fi înțeleasă chiar atunci cînd semnalul vorbirii este perceput cu zgomot. În același timp înțelegerea contextului vorbirii este bazată pe o cunoaștere largă a lumii înconjurătoare, ceea ce ridică probleme serioase din punctul de vedere al recunoașterii vorbirii cu ajutorul calculatoarelor electronice.

Dezvoltarea programelor pe calculator suficient de sofisticate pentru a înțelege vorbirea continuă a unei persoane oarecare este dificilă. Numai atunci cînd programatorii simplifică problema — prin izolarea cuvintelor, limitarea vocabularului sau a numărului de vorbitori prin constrîngerea căilor pe care pot fi formate propoziții — se pot utiliza calculatoarele electronice.

În prezent se comercializează jocuri și jucării care răspund la comenzi date prin voce ; în acest domeniu mai există însă unele dificultăți, dintre care menționăm : restricții ale vocabularului (10—200 cuvinte), necesitatea „școlarizării" noilor vorbitori, necesitatea unei dicții clare, cu pauze între cuvinte și impunerea utilizării unor arii de procesoare cu performanțe superioare, pentru operare în timp real.

### 4.3.4. SISTEME DE VEDERE ARTIFICIALĂ

Oamenii pot vedea și înțelege cu ușurință semnificația unor imagini complexe, spre deosebire de calculatoarele electronice care au performanțe foarte slabe din acest punct de vedere. De exemplu, piesele de pe un conveier sînt recunoscute și localizate imediat, dar numai în condiții limitative și cu o iluminare corespunzătoare.

În ciuda unor progrese înregistrate recent, dezvoltarea unui sistem de vedere realizat cu calculatoare de uz general, apropiat din punctul de vedere al performanțelor de posibilitățile ochiului uman, este încă îndepărtată. Dificultățile sînt legate de realizarea dispozitivelor de imagine — senzori, cum ar fi, de exemplu, vidiconul — care nu oferă informații suficiente pentru o descriere clară a scenei observate, precum și de adîncimea de redare, care se pierde, astfel încît se obțin suprapuneri. O imagine poate apare de culoare neagră datorită reflexiei scăzute, iluminării reduse sau unghiului de vedere nefavorabil. De aici rezultă necesitatea unor cunoștințe vaste privind atît imaginile oferite de diferitele obiecte, cît și modul de combinare a acestora.

Vederea implică o capacitate a memoriei foarte mare și calcule laborioase. Pentru o imagine de $1\,000 \times 1\,000$ pixeli, cele mai simple proceduri cer $10^8$ operații ; retina umană, cu $10^8$ celule care lucrează la 100 Hz, realizează cel puțin 10 miliarde de operații pe secundă, iar cortexul vizual al creierului are posibilități mult mai mari.

Majoritatea sistemelor de vedere se bazează pe un proces ce cuprinde trei etape — de la imaginea brută



pînă la descrierea finală. La început sînt extrase principalele caracteristici din imagine, prin detecția imaginilor și transformarea acestora în linii, curbe sau segmente ; în etapa intermediară se scot informațiile care dau cele trei dimensiuni, cum ar fi orientarea și distanța față de observator, iar în final rezultă o descriere a scenei. În ultima etapă, calculatorul compară imaginea determinată cu modelele memorate ale unor obiecte. Multe sisteme comercializate nu includ toate etapele, pentru a nu se ajunge la prelucrări de complexitate ridicată.

Sistemele de vedere uzuale transformă imaginea cu mai multe trepte de gri în reprezentări binare alb/negru, obținîndu-se și liniile generale ale obiectelor ; acestea sînt măsurate și comparate cu datele din memorie. Dacă se identifică obiectul, poziția și orientarea, atunci nu mai este necesară interpretarea tridimensională.

Firma General Motors Corp. utilizează un sistem de vedere cu calculatorul, denumit *Consight*, aflat în legătură cu roboții de asamblare. Alte sisteme similare realizează inspecții, sortări de piese, ghidarea în sudura cu arc și asistă roboții într-o diversitate de aplicații.

Sisteme mai complexe sînt cerute pentru utilizări speciale — interpretarea fotografiilor și navigația automată —, situații în care vederea nu este limitată la un spațiu închis, iar iluminarea și alți parametri nu sînt controlați cu precizie. În principal s-au depus eforturi deosebite pentru obținerea de informații pe cele trei dimensiuni (3-D) la nivelul intermediar și descrierea mai exactă la nivelul superior.

Determinarea formelor tridimensionale din diferite imagini se face cu ajutorul umbrelor, texturărilor, limitelor lumină/umbră și al altor caracteristici care înseamnă tot atîtea limitări. În prezent este în curs de elaborare o nouă variantă pentru programele de vedere, bazată pe prelucrarea imaginii primare prin extragerea caracteristicilor de constrîngere. Procesul continuă cu reprezentarea matematică a regulilor fizice și geometrice, care sînt semnificative pentru generarea imaginii din scena primară, cum ar fi de exemplu determinarea intensității unui punct al imaginii pornind de la regulile privind

orientarea și reflexia suprafețelor precum și unghiul de iluminare. Folosirea regulilor generale permite extragerea setului de constrîngeri ale scenei primare prin efectuarea unor măsurători asupra imaginii.

Deși nu s-a ajuns încă la elaborarea de programe complete pentru determinarea formelor 3-D din imagini arbitrare, s-au înregistrat succese importante în analiza imaginii unor scene cu limitări date. O asemenea metodă, dezvoltată de B. Horn de la MIT, poate interpreta imagini ale obiectelor netede, dispuse în lumina difuză.

Constrîngerile sînt determinate prin calcule laborioase, fapt ce constituie principalul dezavantaj al metodelor descrise ; sînt necesare calculatoare electronice pentru aplicații specifice, cu posibilitatea de a lucra în paralel, în vederea executării calculelor în timp real. Perspectivele obținerii în timp real a caracteristicilor 3-D din scene arbitrare sînt legate de dezvoltarea actuală a procesoarelor cu $10^{10}$ pînă la $10^{13}$ instrucțiuni pe secundă, deci avînd posibilități similare cu cele ale retinei.

La nivelul superior, recunoașterea și identificarea obiectelor constituie o problemă mult mai complexă, cu progrese mai lente.

Metoda principală pentru recunoașterea obiectelor și descrierea scenelor este vederea pe bază de model, care constă în folosirea unor tipuri de obiecte standard din memorie pentru o anumită scenă, cum ar fi de exemplu pereții, mesele, scaunele etc., în cazul analizei unor camere. Dificultățile constau însă în faptul că obiectele arată altfel dacă sînt privite din unghiuri diferite.

Unele metode au ca punct de plecare corelațiile existente între imaginile memorate și obiectele ca atare.

Sistemul Argos, elaborat de S. Rubin de la Universitatea Carnegie-Mellon, utilizează imagini complete ale unei scene date, ca model de bază, programului revenindu-i sarcini limitate, cum ar fi găsirea unghiului din care se privește. Eroarea este de circa 20%, în cazul acestui sistem, dar aplicațiile sînt limitate, ca de exemplu în coordonarea roboților pentru a se deplasa în spații limitate.



În domeniul celor mai noi metode privind realizarea sistemelor artificiale de vedere, mediul exterior este reprezentat printr-un set de reguli care descriu proprietățile obiectelor sau prin proceduri pe care programul le solicită în anumite situații date. Pentru spații deschise a fost dezvoltat în anul 1980, la Universitatea Kyoto din Japonia, un sistem care mai întîi împarte imaginea în arii de aceeași culoare sau umbră, apoi grupează zonele adiacente pentru a identifica automobile, clădiri, străzi etc.

La Universitatea Stanford, R. Brooks și T. Binford au elaborat, în anul 1981, sistemul Acronym care modelează obiectele tridimensional, sub forma unor cilindri. În baza analizei asupra dimensiunilor și orientării cilindrilor se aproximează diferite obiecte. Metoda este aplicată de firma Hughes Aircraft Co. pentru dezvoltarea unui sistem de interpretare foto destinat monitorizării aeroporturilor.

Descrieri tridimensionale sînt generate și de sistemul Incremental 3-D Mosaic, proiectat la Universitatea Carnegie-Mellon ; se utilizează modele-bloc pentru clădiri, văzute sub unghiuri diferite.

Cercetări reprezentative au mai fost încorporate în proiectul de vedere pentru roboți Hand-Eye dezvoltat de MIT și Universitatea Stanford, sistemul de prelucrare modelată a informațiilor, PIPS (*P*attern-*I*nformation *P*rocessing *S*ystem), apărut în Japonia, precum și în sistemul de interpretare a imaginilor, IUS (*I*mage-*U*nderstanding *S*ystem), destinat armatei.

## 4.4. PROGRAMAREA ROBOȚILOR

Pînă în prezent nu s-a putut stabili cu precizie dacă în antichitate constructorii de automate cunoșteau conceptul de programare mecanică.

Inginerului francez Solomon de Caus (1576—1626) i se atribuie această realizare, folosită la orga automată cu program, dar primele aplicații cunoscute ne-au rămas din secolul al XVII-lea — orologiile cu personaje care apăreau în scene diferite, în funcție de oră.

În secolele următoare tehnica programării mecanice s-a dezvoltat atît pentru orologii, cît și pentru cutii muzicale sau alte automate (de desenat, scris etc.). Dintre realizările deosebite se mai menționează rața lui Vaucanson (sec. al XVIII-lea) care utiliza mecanismul cu came.

Procesul de mecanizare început în secolul al XVIII-lea a continuat cu automatizarea bazată pe puterea mecanică a mașinilor cu vapori și, ulterior, a mașinilor electrice, pînă la introducerea informaticii, la mijlocul secolului al XX-lea. În orice caz, chiar și cele mai complexe automate nu au prezentat nici un fel de inteligență. Pentru a demonstra existența inteligenței mașinile trebuie să-și modifice comportamentul în funcție de schimbările ce survin în mediul exterior.

Primul robot inteligent a fost construit de inginerul spaniol Leonardo Torres Quevedo, în anul 1914, pentru a juca finaluri de șah (rege și turn contra rege). Posibilitățile dispozitivului erau limitate și a trebuit să se aștepte pînă la sfîrșitul anilor '70 pentru a se putea vorbi de inteligența roboților.

Structura conceptuală a robotului cuprinde brațul mecanic, unitatea de alimentare și comandă, precum și sistemul de control. Analiza sistemelor de control folosite în decursul timpului conduce la descrierea evoluției comportării „inteligente" a roboților.

Sistemul de control trebuie să asigure începerea, controlul și încheierea mișcării brațului dintr-o poziție în alta, conform unei secvențe date, memorarea pozițiilor, direcțiilor și secvențelor mișcării, precum și interfața cu senzorii externi. În acest fel se reglează fluxul de energie și alte resurse ale sistemului, pe baza unor instrucțiuni date pentru îndeplinirea unei anumite operații.

La început sistemele de control au fost de natură electro-mecanică, urmate de sisteme electronice cablate. În prezent s-a ajuns la microprocesoare, precum și la mini sau microcalculatoare electronice ; numai roboții de tipul *pick-and-place* nu încorporează tehnica avansată, mulțumindu-se cu sisteme dintre cele mai simple.

Robotul de tip *pick-and-place* este un ansamblu mecanic multiax al cărui terminal poate ocupa numai un



număr limitat de puncte stabile în cursa sa. În general sînt două puncte stabile pe ax deci în total $2^n$ puncte pentru un robot cu $n$ axe. Prin controlul unui singur ax se obțin $2^{n-1}$ curbe, iar în cazul a două axe, $2^{n-2}$ suprafețe în spațiul de lucru al robotului. Cele trei cazuri sînt prezentate, pentru $n=3$, în fig. 4.6. În situația în care

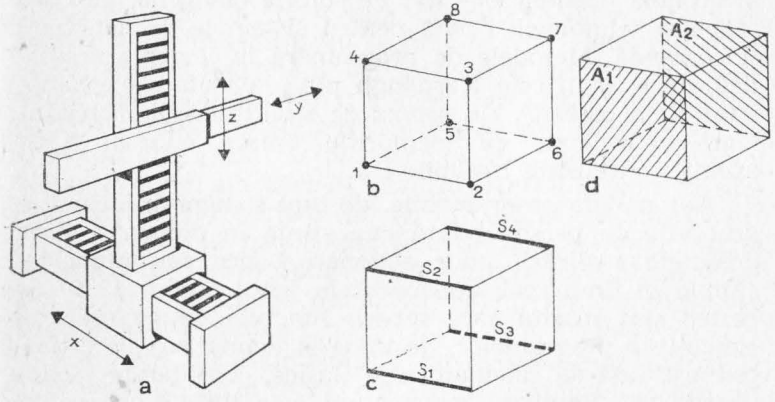



*Fig. 4.6.* Formele descrise de roboții *pick-and-place* :
*a* — robot în coordonate carteziene ; *b* — domeniu format din opt puncte ; *c* — domeniu format din patru segmente (axa *x* comandată) ; *d* — domeniu format din două arii (axele x și y comandate).

pe fiecare ax se stabilesc $m$ poziții, numărul total de puncte devine $m^n$, fiind limitat numai de precizia robotului, obținîndu-se astfel roboți cu mișcare continuă, repetitivă *(play-back)*. Metoda menționată nu mai este însă folosită din cauza apariției și dezvoltării tehnicii de calcul și memoriilor electronice.

În cazul roboților comandați de automate programabile sau de calculatoare electronice se pune problema utilizării unui limbaj de programare care, de fapt, nu este specific mașinii. Pentru tipurile menționate se poate vorbi de un limbaj în codul mașinii, în timp ce pentru roboții *pick-and-place* comandați de logică electrică realizată prin cablare sau pneumatică limbajul este asociat tehnologiei de comandă. Entitatea tehnologică de bază în acest caz este memoria binară, fără a fi necesară pre-

zența unei unități aritmetice și logice. Se menționează că în industrie se mai utilizează mulți roboți cu relee electromecanice, pentru care logica de comandă este, de asemenea, cablată.

După 1976, logica pneumatică a luat un avînt considerabil în echipamentele de automatizare ; unul din avantajele acesteia este dat de soluția omogenă, pe baza aceleiași tehnologii, găsită pentru sistemele de acționare și comandă. Metodele de programare în logica pneumatică, de la matricele Karnaugh pînă la metodele geometrice și în cascadă, țin seama de similitudinea distribuitoarelor utilizate cu memoriile binare, cum ar fi de exemplu circuitele bistabile RS.

Automatele programabile, de fapt sisteme informatice deservite de personal fără cunoștințe în domeniu, fiind destinate realizării unor acționări logice sau numerice simple în timp real, au apărut în jurul anului 1970. Acestea sînt mașini care separă funcțiile de generare și execuție a programelor, de unde și denumirea de biprocesoare, față de calculatoarele clasice, considerate monoprocesoare. Setul de instrucțiuni este limitat la cca 20 instrucțiuni față de 60 la microprocesoare și 250 la calculatoare. Problemele în timp real sînt tratate de automatele programabile în mod sincron, fără întreruperi, față de modul asincron utilizat de calculatoarele electronice.

Funcțiile logice și de calcul numeric sînt cele mai potrivite pentru comanda manipulatoarelor *pick-and-place*. Ca metode de lucru se utilizează fie limbajul cu relee și reprezentare grafică sub formă de diagrame Ladder, fie o reprezentare de tip booleană sau numerică.

Majoritatea roboților *play-back* se programează prin instruire. Limbajele de nivel înalt și-au făcut apariția la început în cazul roboților destinați realizării operațiilor de asamblare. Pentru producător ambele metode de programare, prin limbaj de nivel înalt și prin instruire, sînt dificile, în timp ce pentru utilizator sînt relativ simple.

Proiectanții se străduiesc să asigure și pentru roboții *pick-and-place* aceleași posibilități de programare. Dispozitivul de instruire cuplat la automat permite realizarea de către manipulator a ciclului dorit, executat direct de operator. După instruire, cutia se deconectează și ciclul poate fi repetat în mod automat la infinit de către robot.



Roboții simpli, bazați în general pe elemente modulare, pot fi adaptați să efectueze diverse sarcini prin reglarea cursului lor și programării ciclului lor de funcționare. Astfel de roboți sînt utilizați în operații simple pentru care flexibilitatea solicitată de ansamblu este slabă, ca de exemplu la încărcarea/descărcarea unei prese sau mașini-unelte ; în acest caz robotul simplu deservește numai mașina respectivă și flexibilitatea sa se limitează la gama de piese pe care o poate manipula. Roboții simpli nu sînt în general utilizabili pentru sarcini diferite sau pentru aceeași sarcină dar pe mașini diferite. Interesul manifestat față de acești roboți constă în structura lor modulară care permite asamblarea simplă și un cost redus datorită elementelor standard din care sînt construiți.

Evoluția roboților simpli este legată de adaptarea la fabricația de serie mare, caz în care intervin modificări puține, rezultînd o durată lungă de viață. În ciuda simplității, instalarea roboților descriși mai sus este de durată. Se au în vedere probleme ca descompunerea sarcinilor și alegerea reperelor, programarea logică prin metode de cablare sau automate și punerea în funcțiune, ajungîndu-se la intervale de la cîteva luni pînă la un an.

Evoluția costurilor orare, $C$, depinde de costul robotului, $C_r$ inclusiv valoarea reutilizabilă la sfîrșitul aplicației, $\alpha_r$ (%), costul proiectului, $C_p$, durata între două modificări, $T_m$, costul modificării, $C_m$, costul de funcționare pe oră, $C_f$, și durata totală de funcționare, $T_t$, după următoarea formulă :

$$C = \frac{(1-\alpha_r)C_r + C_p}{T_t} + \frac{C_m}{T_m} + C_f.$$

Pe baza acestei formule se pot compara producția manuală cu cea automată rigidă și, respectiv, flexibilă, rezultînd situația din tabelul 4.5.

TABELUL 4.5. **Evoluția costurilor în cazul producției manuale și automatizate**

| Parametrul / Producția | $C_r$ | $\alpha_r$ | $C_p$ | $T_m$ | $C_m$ | $C_f$ | $T_i$ |
|---|---|---|---|---|---|---|---|
| Manuală | f. mic | f. mic | mediu | impor-tant | f. mic | mare | impor-tant |
| Automată — rigidă | mediu | mic | mare | mare | mare | mediu | mare |
| Automată — flexibilă | mare | mare | mediu | impor-tant | mic | mediu | impor-tant |

Spre deosebire de roboții simpli, la care capătul cursei poate fi stabilit numai mecanic, roboții comandați sînt programabili pe toată cursa pe fiecare din gradele de libertate. Roboții pot fi însă realizați în tehnologie mixtă, unele grade de libertate fiind comandate, iar altele avînd două poziții. Dacă cel puțin un grad de libertate este comandat, atunci roboții respectivi sînt considerați comandați.

Roboții comandați pot fi de tipul punct cu punct, cu traiectorie calculată, cu traiectorie înregistrată și, respectiv cu traiectorie programată.

La roboții punct cu punct sînt înregistrate în memorie numai cîteva puncte semnificative ale traiectoriei, precum și condițiile de trecere dintr-un punct la altul. În timpul execuției, robotul trece de la un punct la altul prin transmiterea informației asupra poziției finale a tuturor axelor, ceea ce conduce de cele mai multe ori la traiectorii imprevizibile, timpul de parcurgere fiind diferit. Rezultă, deci, că programul unui robot punct cu punct este constituit dintr-un număr de poziții și condițiile de declanșare a mișcării spre punctul următor, acestea fiind de obicei exprimate prin intermediul algebrei booleene sau al valorii reale a semnalului. Toate variabilele care privesc un punct al programului sînt memorate într-un bloc de date a cărui lungime depinde de caracteristicile robotului — numărul de axe, rezoluția, complexitatea etc. —, dar nu depășește cu mult 20—25



cuvinte de 16 biți. Numărul de puncte din program este limitat de capacitatea memoriei, fiind mai mare doar în cazuri speciale cum ar fi de exemplu roboții pentru sudură. Timpul de deplasare între două puncte depinde de viteza brațului robotului, fiind de ordinul secundelor, de aici rezultînd posibilitatea utilizării chiar și a memoriilor mai lente. Cu toate acestea, în ultimul timp se folosește memoria centrală pentru facilitarea editării programelor, deoarece se pot introduce puncte noi fără a mai fi necesară recopierea programului. Ca aplicații tipice se menționează încărcarea/descărcarea reperelor și sudura punct cu punct.

Roboții cu traiectorie calculată diferă de cei de mai sus prin faptul că traiectoria este impusă de programator. Un caz tipic de aplicație este sudura cu arc, la care se impune viteza de deplasare. Constructiv, roboții cu traiectorie calculată au complexitatea ceva mai ridicată, dar informațiile din memorie sînt practic similare cu ale roboților punct cu punct, cu excepția datelor privind tipul traiectoriei și vitezei de deplasare care înseamnă un cuvînt de 16 biți.

Roboții cu traiectorie înregistrată trebuie să urmărească mișcări complexe care să copieze mișcările efectuate de experți umani. Aplicația tipică este vopsirea. Repetabilitatea se poate obține prin înregistrări analogice, dar această modalitate ar conduce la un volum foarte mare de informații necesare ; în practică se utilizează tehnici digitale prin eșantionarea traiectoriei impuse. Pentru a eșantiona o traiectorie la fiecare milimetru, dacă viteza este de 1 m/s, rezultă o frecvență de 1 kHz. Avînd în vedere lungimea unui program pentru roboții cu traiectorie continuă, se utilizează memoriile de masă (discuri sau bandă magnetică, dischete etc.), transferul în memoria centrală realizîndu-se prin intermediul memoriilor tampon.

În cazul roboților cu traiectorie programată, traseul nu este înregistrat prin instruire, ci este programat prin limbaje simbolice cu ajutoul calculatoarelor electronice.

Transformarea informațiilor din memorie în comenzi pentru sistemul mecanic se face prin intermediul unei unități de comandă. Roboții punct cu punct, fiind prin-

tre primii roboți comandați, posedau o unitate de comandă cablată cu memorie asociată realizată cu circuite logice. Apariția microprocesoarelor a determinat o schimbare radicală ; la ora actuală toți roboții comandați încorporează unul sau mai multe microprocesoare cu logică sau programe dezvoltate de proiectantul robotului pentru a îndeplini sarcinile cerute. Partea electronică a robotului realizează instruirea, gestiunea de programe și executarea programelor.

În fig. 4.7. se prezintă schema bloc a sistemului informatic al roboților.

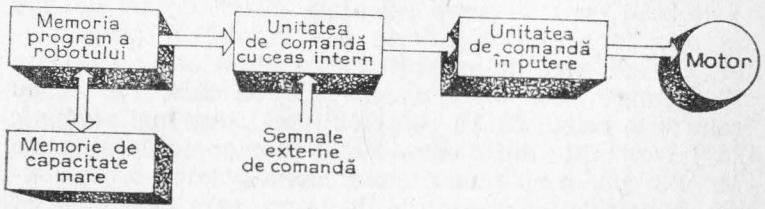

Memoria program a robotului
Unitatea de comandă cu ceas intern
Unitatea de comandă în putere
Motor
Memorie de capacitate mare
Semnale externe de comandă

*Fig. 4.7.* Schema-bloc a sistemului informatic al roboților.

Funcțiile pe care trebuie să le îndeplinească unitatea de comandă sînt următoarele : urmărirea secvenței instrucțiunilor, calculul traiectoriei, transformarea de coordonate și comanda efectivă.

Programarea prin instruire reprezintă metoda cea mai utilizată pentru roboții instalați, de la primii roboți de tip Unimate pînă la cei realizați în prezent. Alte metode se utilizează numai pentru aplicații de mare complexitate sau în cazul în care timpul de instruire este prea lung ori nu se justifică — seria prea mică de fabricație nu permite amortizarea rapidă sau generarea automată de programe este mai simplă.

Metodele de programare prin instruire necesită efectuarea unei demonstrații fizice de către un operator expert, lucru care se poate realiza prin modalități diferite.

Roboții punct cu punct și cei cu traiectorie calculată impun deplasarea roboților după traseul care trebuie înregistrat de către operator fie manual, fie prin



telecomandă. De asemenea operatorul trebuie să transmită unității de comandă și alte informații, cum ar fi punctul de înregistrat, natura acestuia (oprire sau trecere), semnalele externe pentru așteptare sau declanșare, temporizarea, viteza de deplasare, natura traiectoriei etc.

Inițial toate aceste informații erau introduse cu ajutorul pupitrelor de comandă, dar pe măsura creșterii numărului datelor de intrare s-a apelat la echipamentele de tehnică de calcul. În acest fel programul care coordonează activitatea robotului poate să ceară datele în clar sau codificat, funcție de cum s-a stabilit în prealabil. Posibilitățile de editare de programe pot fi utilizate, de asemenea, în faza de instruire, de exemplu pentru a face unele corecturi de traiectorie. Roboții mai sofisticați dispun și de posibilitatea de a genera în mod automat o secvență de mișcări, plecînd de la una singură.

Pentru trasee mai complexe, de pildă în cazul roboților cu traiectorie înregistrată, instruirea se face numai prin prezentarea traiectoriei, deși în acest caz roboții au o greutate apreciabilă. Există mai multe metode de instruire, de la deplasarea manuală asistată sau nu în pilotare, repetarea cu un mecanism simplu similar robotului, pînă la telecomandă ; pentru acești roboți sînt necesare comenzi suplimentare pentru elaborarea programelor. De obicei pentru un program optim sînt necesare mai multe încercări pentru a se ajunge la mișcări corecte și la un timp minim de deplasare.

La mijlocul anilor '60 au apărut atît roboții mobili, cît și cei simpli, sub forma unor brațe manipulatoare capabile să acționeze funcție de diversitatea condițiilor exterioare. Din grupa roboților mobili se menționează robotul Shakey, iar din cea a roboților simpli brațele de tipul Rancho, Stanford și MIT, ultimele două fiind realizate și comercializate de către firma VICARM.

Mijloacele oferite de informatică au început să fie utilizate pentru realizarea roboților cu performanțe superioare începînd de la mijlocul anilor '70, cînd se poate vorbi deja de cea de-a doua generație de roboți, cu atît mai mult cu cît puterea de calcul în echipamentele electronice se mărește semnificativ după apariția și dezvol-

tarea microprocesoarelor (1971). În acest fel încep să fie experimentate limbajele de programare în comanda și controlul roboților. Dintre primii roboți care folosesc tehnica programării se menționează SIGMA și PUMA de la firmele Olivetti și, respectiv, Unimation (1977).

Introducerea pe scară largă a celei de-a doua generații de roboți s-a produs însă abia în jurul anului 1982, cînd existau deja numeroase limbaje de programare pentru roboți — AML (elaborat de firma IBM), LPR (SCEMI), SERF (SANKYO) etc. — ca și firme specializate în realizarea comenzilor electronice pentru roboți programabili, ca de exemplu General Electric ((limbajul PCL) și GIXI — filiala firmei CEA — (limbajul ROL).

Apariția și folosirea microprocesoarelor au permis accesul utilizatorilor la puterea de calcul, ceea ce a condus la extinderea considerabilă a aplicațiilor roboților. Posibilitățile microprocesoarelor pot fi puse cel mai bine în evidență prin programarea în limbajul mașinii sau în limbajul de asamblare ; de aici rezultă performanțe superioare care se întîlnesc în principal la roboții specializați. Dezavantajul acestei metode constă în complexitatea programării, astfel încît metoda nu este la îndemîna tuturor utilizatorilor.

Așa cum în tehnica de calcul s-a găsit o metodă simplificatoare — compilarea, adică traducerea limbajelor de nivel înalt în limbajul mașinii —, în robotică s-a ales interpretarea pentru decodificarea programului utilizatorului și comanda acțiunilor roboților.

În funcție de puterea setului de instrucțiuni pus la dispoziția programatorului, limbajele pot fi de nivelul 1 — similar comenzilor numerice pentru mașini-unelte, nivelul 2 — cînd se programează explicit deplasarea robotului, nivelul 3 — programarea implicită prin modelarea pieselor obiect și nivelul 4 — în curs de cercetare — ce implică folosirea inteligenței artificiale deoarece se indică numai obiectivul final fără a se preciza etapele intermediare ale mișcării sau condițiile din mediul exterior.

Producătorii de roboți trebuie să pună la dispoziția utilizatorilor limbaje care să cuprindă programarea generală, independentă de tipul robotului, metodologia de



modelare geometrică a spațiului, instrucțiunile pentru deplasare, precum și modul de folosire a senzorilor și sculelor de lucru specifice acestor dispozitive complexe.

În general, programarea prin limbaj se face off-line, adică fără utilizarea robotului sau a sistemului său de control. Programarea și traducerea se realizează, în acest caz, pe un sistem de calcul distinct, principala problemă de rezolvat fiind definirea codului intermediar de lucru care ar trebui să fie comun pentru toate categoriile de roboți. Avînd în vedere specificul manipulării, s-au propus mai multe coduri intermediare dintre care se menționează IRDATA. Evoluția roboticii face însă practic imposibilă definirea tuturor instrucțiunilor de bază care să fie valabile pentru diferitele clase de roboți.

Dintre limbajele de programare se menționează ALGOL, PL1 și APL apărute în anii '60, PASCAL și BASIC în anii '70 și, mai recent, ADA.

Modelarea geometrică a spațiului este necesară deoarece pozițiile robotului se calculează, spre deosebire de tehnica instruirii unde ele sînt cunoscute din deplasările efectuate de axe. Modelarea se face, de obicei, prin variabile care definesc poziția și orientarea obiectelor în spațiul ales. De cele mai multe ori diferențele existente între obiecte se apreciază pe baza unor repere specifice, iar poziția în spațiu prin transformări între referențialul de bază și cel al obiectului.

Deplasarea robotului se face fie prin indicarea pozițiilor absolute, fie a deplasărilor relative față de poziția curentă.

Senzorii și sculele determină acțiunea robotului în mediul exterior prin deplasări de tip punct cu punct sau continuu.

În cele ce urmează se face o scurtă trecere în revistă a principalelor limbaje de programare pentru roboți.

• *AL (Arm Language)* este limbajul care cunoaște o dezvoltare foarte importantă. Realizat de cercetătorii de la Stanford, limbajul AL a fost îmbunătățit permanent în perioada 1974—1983. Prima sa versiune era consti-

tuită dintr-un model de planificare pentru compilarea programelor pe microcalculatoare PDP-10 și generarea traiectoriilor. Un cod intermediar a fost executat pe microcalculatorul PDP 11/45. Limbajul AL a fost difuzat în întreaga lume, două versiuni complete ale sale fiind implementate la universitățile din Karlsruhe și Tokyo. Limbajul AL este de tip ALGOL și poate fi utilizat pe roboți VICARM și PUMA.

• *AML* (*A Manufacturing Language*) este limbajul lansat pe piață, după cercetări efectuate în perioada 1977—1982, de firma IBM pentru a fi folosit de robotul hidraulic cartezian RS-1 sau 7565 fabricat de IBM și robotul electric 7535, din gama SCARA, fabricat de firma Sankyo din Japonia. Limbajul AML, de tip APL, și realizarea roboților cu structură carteziană au determinat, la mijlocul anilor '70, amplificarea cercetărilor în domeniu. În consecință au fost dezvoltate și experimentate noi limbaje — MAPLE, EMILY, AUTOPASS — care apoi, începînd din anul 1979, au fost folosite de cîțiva roboți produși de IBM. De altfel trebuie să precizăm faptul că firma IBM este primul constructor de roboți care a prezentat un echipament standard — robotul RS-1 asistat de un minicalculator din seria 1. Versiunea limbajului AML folosită de roboții produși de Sankyo este un subansamblu AML : programele sînt elaborate pentru un microcalculator IBM-PC, fiind adaptate ulterior la specificul echipamentelor realizate de firma japoneză.

• *IRL* (*Intuitive Robot Language*) este un limbaj similar comenzilor numerice pe calculator, dezvoltat în perioada 1978—1980 de firma elvețiană Microrobot pentru gama sa de roboți de asamblare de înaltă precizie. Această gamă este compusă din modelul electric Souris, cu șase grade de libertate, modelul hidraulic Castor și modelul Ecureuil, lansat în 1983. Acești roboți au fost utilizați exclusiv în industria de ceasuri, dar din 1983 au fost livrați și pe piață.

• *LM* (*Language de Manipulation*), de tip PASCAL, a fost lansat de laboratorul de robotică de la Universitatea din Grenoble, în perioada 1979—1983. Inspirat de limbajul AL, acesta cuprinde principalele sale caracteristici — mai puțin programarea paralelă —, fiind însă



implementat pe microcalculatoare LSI-11/23 și 68 000 pentru roboți de tip SCEMI și de alte tipuri.

• *LPR* (*Langage de Programmation de Robot*) a fost dezvoltat de RNUR-DTAA, în perioada 1978—1983, pentru comanda roboților ACMA-Renault ; este cunoscut și sub denumirea de comandă versiunea 5 sau V5. Comanda anterioară, V3, utiliza minicalculatoare MITRA 105 sau Philips P-850 programate în limbaj asamblor, într-un mod specific pentru fiecare clasă de aplicații — în principal de tip punct cu punct. Pentru a răspunde la diferite cerințe, a fost dezvoltată o nouă structură informatică pentru microprocesoare de 16 biți (8086). Pentru a permite dezvoltarea unor aplicații complexe, limbajul LPR a fost conceput pe baza GRAFCET (metodologie franceză de descriere a caietului de sarcini a unui automat logic). În această formulare dezvoltarea se face în codul microprocesoarelor, cu minicalculatoare Philips, MITRA sau VAX. La început limbajul LPR a fost folosit de Renault pentru propriile necesități ale firmei, fără ca alți utilizatori să aibă acces la el ; această concepție a fost însă abandonată nu de mult.

• *MCL* a fost conceput în cadrul proiectului de fabricație integrată asistată de calculator ICAM (*Integrated Computer Aided Manufacturing*), pentru a unifica programarea roboților. Proiectul este realizat de filiala Mc AUTO a firmei Mc Donnel Douglas, care oferă deja pe piață versiunea MCL/11 interactivă pe microcalculator PDP-11, față de versiunea inițială pentru IBM-370. MCL — o extindere a limbajului APT — a fost elaborat în scopul de a permite programarea celulelor flexibile, adică un ansamblu de mașini deservite de unul sau mai mulți roboți. În prezent MCL poate controla roboții Cincinnati T3 și Westinghouse Allegro.

• *OMSI* — o versiune interactivă a limbajului AL — este realizat în PASCAL pentru microcalculatorul PDP-11/45. Acest limbaj poate comanda roboții de tip PUMA.

• *PLAW* (*Programming Language for Arc Welding*) este un limbaj pentru programarea sudurii cu arc. PLAW este adaptat la sudura „inteligentă" pentru a depăși tehnica de programare prin instruire. Limbajul a

fost dezvoltat de firma Komatsu pentru roboții săi cartezieni din seria RW echipați în mod corespunzător.

• *POINTY* este un limbaj similar cu AL, ce poate fi executat pe microcalculatoare din familia PDP-11 pentru a oferi o versiune mai interactivă la nivelul programării.

• *RAIL*, conceput de firma AUTOMATIX în perioada 1979—1982, este primul limbaj de robotică destinat atît rezolvării problemelor de manipulare, cît și a celor de vedere. Bazat pe limbajul PASCAL, RAIL este programat și se execută pe o mașină realizată de Automatix pe baza microprocesorului 68 000. Aplicațiile de robotică privesc în principal sudura continuă pe roboți de tip Hitachi și DEA.

• *ROL (Robot Language)* a apărut ca rezultat al activității de cercetare pentru elaborarea limbajului LAMA-S, în cadrul proiectului Spartacus care urmărea lansarea pe piață a unei comenzi complete — echipamente și logică — ce putea fi adaptată la toți roboții. Sistemul, lansat în 1983, este modular și poate fi folosit de la o configurație foarte simplă, în două axe, pînă la o configurație pentru mai mulți roboți, mai multe mașini și captatori evoluați. Programarea se efectuează pe microcalculatoare IBM-PC.

• *SERF (Sankyo Easy Robot Formula)* a fost realizat în perioada 1977—1979 de firma Sankyo pentru roboții săi instruibili și este derivat din lucrările întreprinse în cadrul proiectului SCARA. Roboții Sankyo fiind orientați spre sarcini de mici asamblări în care paletizarea joacă un rol important, s-a ales pentru prima dată în Japonia un limbaj textual. Limbajul este de nivel elementar, în stilul limbajelor de comenzi numerice. Dezvoltarea programului de aplicații se face pe o consolă specializată și pe baza microprocesorului Z-80.

• *SIGLA (SIGma Language)* — primul limbaj disponibil comercial pentru un robot industrial — a fost cercetat, în perioada 1974—1976, de firma Olivetti pentru roboții săi SIGMA cu structură carteziană. SIGLA este un limbaj simplu, inspirat din limbajele de comenzi numerice. Roboții SIGMA sînt orientați spre aplicații de asamblare, mic uzinaj și sudură ; acești roboți sînt comercializați de firma Westinghouse, în paralel cu roboții PUMA, de unde posibilitatea ca SIGLA să intre în competiție cu limbajul VAL.



• *VAL (Variable Assembly Language)*, limbaj de tip BASIC, a fost cercetat în perioada 1979—1983 la Stanford, plecînd de la limbajul WAVE, pentru mici roboți destinați activității de cercetare. În continuare limbajul VAL a fost dezvoltat în cadrul proiectului PUMA, recent fiind adoptat pentru toți roboții de tip Unimate. Versiunea VAL II de tip PASCAL oferă numeroase avantaje față de varianta inițială, fără a aduce însă modificări importante, fiind executabilă pe minicalculatoare LSI-11/23 și microprocesoare 6502.

În prezent se depun eforturi deosebite în vederea îmbunătățirii limbajelor actuale și dezvoltarea unor limbaje noi care să lărgească gama aplicațiilor pentru roboți.

## 4.5. APLICAȚII ALE ROBOȚILOR INDUSTRIALI

Roboții sînt întotdeauna comparați cu omul în ceea ce privește caracteristicile și comportamentul, deși nici un robot nu a atins pînă în prezent posibilitățile de învățare și judecată, apreciere artistică, reproducere etc, specifice ființelor umane. Cu toate acestea, prin eforturile specialiștilor în robotică — domeniu în care se implementează cele mai avansate cunoștințe dintr-o serie de discipline diferite, cum ar fi cinematica, dinamica, proiectarea servomecanismelor, puterea fluidelor, electronica digitală și analogică, structura și software-ul calculatoarelor, proiectarea circuitelor integrate, cibernetică, automatica, controlul numeric, ingineria sistemelor, proiectarea angrenajelor, ingineria structurală, tribologia, metalurgia, instrumente senzoriale, recunoașterea caracterelor, ingineria industrială și a fabricației, fiziologia, bionica, psihologia, sociologia, economia, futurologia etc. — deosebirile dintre om și robot se atenuează continuu.

Principalele caracteristici ale roboților realizați pînă în prezent pot fi sintetizate după cum urmează :

- comanda în spațiul de lucru cu articulații cu șase grade de libertate ;
- posibilități de învățare și repetare ;
- memorii locale sau biblioteci de memorii ;
- selectarea din exterior aleatoare a programelor ;
- repetabilitatea preciziei de poziționare ;
- manipularea unor greutăți mari ;
- controlul punct cu punct și continuu ;
- sincronizarea în mișcare cu piesele în lucru ;
- posibilitatea interfațării cu calculatoarele electronice ;
- posibilitatea paletizării și depaletizării ;
- fiabilitate mărită ;
- prețuri care justifică din punct de vedere economic investițiile.

După cum se va arăta în cele ce urmează, roboții au fost deja implementați în majoritatea sectoarelor de activitate pentru a înlocui munca umană, în scopul creșterii productivității și eficienței etc. Aplicațiile roboților sînt prezentate, în principal, pe domenii de utilizare.

### 4.5.1. SUDURA PRIN PUNCTE

Sudura constă în fuzionarea metalelor supuse acestui proces, spre deosebire de lipire unde se utilizează și un alt metal — plumb, cositor sau argint —, cu punctul de topire mai coborît. Cea mai folosită este metoda de sudură prin puncte obținute la trecerea unor curenți electrici mari prin metalele cu care se lucrează ; metoda nu poate fi însă aplicată unor suprafețe mari deoarece căldura se disipă, făcînd imposibilă creșterea temperaturii pînă la limita necesară.

Dispozitivele care realizează sudura prin puncte au fost automatizate în mare măsură ; un dezavantaj îl reprezintă însă necesitatea modificării soluției utilizate cînd pe liniile de fabricație se introduc simultan mai multe tipuri de produse, ca de exemplu în industria de automobile. În consecință, roboții se folosesc în principalele aplicații din acest domeniu. Realizarea echipamentelor



necesare este însă destul de complexă, întrucît articulațiile trebuie să posede pînă la șase grade de libertate.

Primul pas în robotizarea procesului de fabricare a automobilelor a fost făcut în anul 1966, prin conectarea sistemelor de control ale mașinilor de sudură la semnale externe care comandau operațiile de deplasare, sudură, răcire etc. În anul 1969 firma General Motors, urmată după numai un an de firma Daimler Benz, a instalat o linie de 26 roboți pentru sudura prin puncte a caroseriilor. Capacitatea medie de producție a unei linii este de 80 caroserii pe oră. Toți marii producători de autoturisme utilizează masiv roboții în producție, unele firme — Volkswagen și Renault — trecînd la construcția unor modele proprii. Una din cele mai complexe linii de sudură din lume a fost instalată pentru firma Volvo, în Suedia, productivitatea fiind de 50 caroserii pe oră pentru modele de autoturisme cu două și respectiv patru uși.

Tehnica folosită în domeniul auto constă în sudura prin puncte a unui corp asamblat inițial prin simpla prindere a părților componente, operație realizată cu ajutorul roboților. Un sistem de calcul supraveghează alocarea ariei de memorie specifică diferitelor tipuri de produse.

În fig. 4.8 se prezintă schema liniei de producție din uzina de asamblare din Newark a firmei Chrysler ; ea cuprinde șapte posturi de sudură realizate cu 13 roboți de sudură.

Într-o asemenea linie timpul pentru ciclul complet de sudură poate fi determinat luîndu-se în calcul următoarele mișcări : deplasarea pînă la primul post de sudură, reglarea capului de sudură, execuția unui grup de suduri în mod secvențial, deplasarea spre al doilea grup de suduri, modificarea planului de orientare, retragerea capului de sudură, precum și deplasarea de la postul de sudură.

Controlul unui grup de roboți dintr-o linie de asamblare poate fi centralizat cu ajutorul calculatorului, folosind chiar un alt sistem adecvat din întreprinderea respectivă dar cu programe specifice în funcție de operațiile ce urmează a fi executate. Calculatorul contro-

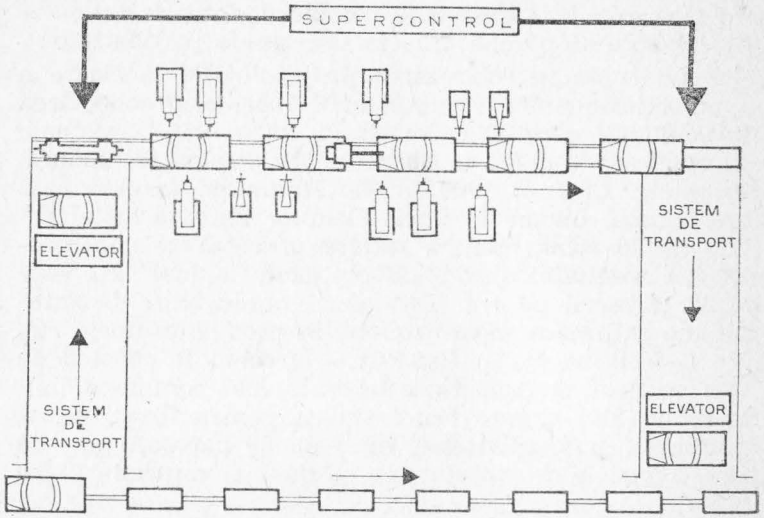



*Fig. 4.8.* Linie de sudură cu roboți a autoturismelor Chrysler.

lează întregul sistem și, în paralel, înregistrează datele statistice privind producția, precum și verificarea calității sudurilor.

### 4.5.2. TURNAREA

În operația de turnare în matrițe, piesele sînt formate prin injecția sub presiune a unor metale neferoase în matrițe metalice. Cele mai utilizate sînt aliajele conținînd aluminiu, zinc, magneziu, cupru și alamă. Instalația de turnare cuprinde o serie de dispozitive pentru prinderea, închiderea sub presiune și eliberarea matrițelor, constituite în mod uzual din două părți. După turnare, piesele sînt de cele mai multe ori răcite, fiind apoi supuse prelucrării pentru îndepărtarea excesului de material.

Turnarea se poate face la rece sau la cald, în funcție de metalul folosit, gradul de automatizare în cazul celor două metode fiind de 45% și, respectiv, 75%. Una din operațiile care consumă mult timp este întreținerea și curățirea matriței, mai ales atunci cînd în acest scop este folosită munca omului.



În procesul de turnare în matrițe roboții au fost introduși încă din anul 1961, acumulîndu-se pînă în prezent un număr impresionant de ore de operare — de ordinul zecilor de milioane. Avantajul principal rezultă din faptul că roboții pot lucra practic în permanență. Pot fi deservite de roboți atît instalațiile de turnare la cald, cît și la rece. Operațiile de turnare, scoatere din matriță ,răcire și finisare a pieselor turnate sînt de tipul „pick-and-place", impunîndu-se roboții din această categorie.

În turnătorii roboții execută, în principal, următoarele operații : descărcarea unei mașini, răcirea, separarea pieselor sau plasarea acestora pe un conveier ; descărcarea alternativă a două mașini ; descărcarea unei mașini, răcirea și finisarea pieselor ; descărcarea unei mașini și pregătirea matriței pentru ciclul următor ; inserția unor elemente suplimentare în matriță și descărcarea mașinii etc.

Pentru înlocuirea operatorului uman cu un robot sînt necesare modificări nesemnificative ale instalațiilor și echipamentelor existente în turnătorii, dar trebuie avute în vedere o serie de elemente dintre care cele mai importante sînt precizia orientării pieselor și complexitatea acestora.

### 4.5.3. SUDURA CU ARC

Procesul de sudură cu arc electric este utilizat pentru îmbinarea electrică a unor suprafețe conductoare. La temperatura înaltă, de arc, ce se obține între două terminale — electrodul de lucru și cel de sudură — se formează o zonă de metal topit care prin răcire conduce la realizarea efectului dorit. Electrodul de sudură este de fapt o sîrmă din metal, alimentată într-un ritm corelat cu consumul necesar pentru întreținerea operației. Pornind de la cărbune sau sîrmă de metal acoperită cu

un material de flux, s-au dezvoltat tehnici de sudură cu electrod de tungsten sau metal în gaz inert pentru aluminiu și aliajele sale, cupru, magneziu, oțel inoxidabil etc. Instalațiile moderne conțin un tub flexibil pentru gaz prin care electrodul metalic trece cu viteză controlată.

Operatorul realizează un scurtcircuit între cei doi electrozi pentru a iniția temperatura înaltă, apoi retrage electrodul de lucru la o mică distanță în vederea generării arcului electric. Poziționarea electrodului față de material și urmărirea traseului pe care se efectuează sudura reprezintă cerințe deosebit de importante.

Procedura descrisă este tracasantă dacă cel care o urmează este omul, rezultînd erori în execuție. De asemenea arcul electric creează condiții neplăcute de lucru : radiații ultraviolete — periculoase pentru ochi —, scînteile, fumul etc. Prin utilizarea roboților aceste dezavantaje sînt eliminate, obținîndu-se în mod repetat suduri de calitate.

Metoda cea mai utilizată este învățarea de către robot a unei secvențe de sudură executate de un operator expert, astfel încît toate informațiile de poziție vor fi înregistrate electronic în program. Ulterior, robotul va repeta cu precizie operațiile învățate. Acest mod de lucru este cunoscut sub denumirea de programare în timp real.

Programarea robotului poate fi realizată și prin metoda punct cu punct, care permite calculatorului să interpoleze cu o bună aproximație parcursul între puncte consecutive.

În cazul sudurii cu arc roboții trebuie să posede pînă la cinci articulații.

Secvențele de lucru furnizate de circuitele de control pentru roboții destinați sudurii cu arc sînt următoarele : alimentarea cu gaz inert ; sudura propriu-zisă ; oprirea alimentării sîrmei și deci stingerea arcului ; întreruperea curentului electric și, ulterior, a gazului inert pentru eliminarea posibilității oxidării suprafețelor în aer datorită temperaturii înalte ; schimbarea poziției și reluarea operațiilor de mai sus.



Pe lîngă sudura cu arc electric se poate realiza și tăierea cu flacără, utilizînd roboți printr-o procedură similară. Precizia rezultată este mai mare decît în cazul folosirii metodelor uzuale, iar programarea nu este diferită.

### 4.5.4. FORJAREA

Prin forjare obiectele de metal sînt prelucrate prin lovituri de ciocan sau prese, după ce mai întîi au fost încălzite pînă la temperatura de plastifiere. De la cunoscutul fierar, prin mecanizare, forjarea s-a diversificat, această operație fiind executată în prezent atît prin lovituri, cît și prin presare, rulare sau alte forme specifice.

Forjarea prin lovituri cu ciocanul folosind matrițe deschise permite obținerea formelor dorite, care de obicei sînt relativ simple ; se utilizează ciocane de pînă la 50 tone, impactul rezultat solicitînd mașini și fundații solide. În cazul formelor complexe ciocanele sînt înlocuite, cu matrițe avînd două secțiuni — superioară și respectiv inferioară —, cu care se aplică serii de pînă la 300 lovituri pe minut. Presele sînt utilizate pentru forjarea unor lingouri de peste 200 tone ; se precizează că, spre deosebire de presarea obișnuită, materialul de lucru se află la o temperatură înaltă. Forjarea prin laminare este un proces special, obținîndu-se un număr limitat de forme. O altă operație specială de forjare se întîlnește în cazul producției de bolțuri și piulițe.

După cum se cunoaște, mediul de lucru în secțiile de forjare este dintre cele mai grele și nocive datorită zgomotului, căldurii excesive și aerului poluat ; în plus operațiile sînt repetitive, deci neatractive pentru personalul muncitor. Ca urmare, s-a impus cu putere introducerea automatizării, roboții găsindu-și la rîndul lor un cîmp larg de acțiune. Datorită însă unor cerințe specifice, cum ar fi viteza mare — necesară pentru a nu se răci piesele în lucru —, modificarea forței de lovire pe parcursul realizării unui produs, precum și posibilitatea apariției unor inconveniente, legate în principal de rămînerea piesei finite în matriță sau transformarea formei

lingoului într-o asemenea măsură încît trebuie folosite la aceeași operație mai multe sisteme de apucare, în forjare se utilizează mai mult manipulatoare comandate sau roboți controlați de un operator uman.

Numărul roboților utilizați la forjare este în continuă creștere deoarece pot lucra cu piese din metal la temperaturi înalte și rezistă perioade lungi de timp în medii foarte ostile. Aplicațiile practice constau în transferul preformelor de metal fierbinte de la forje la instalațiile de tragere, fabricația de lanțuri și alte operații complexe și dificile.

### 4.5.5. DESERVIREA MAȘINILOR-UNELTE

Obținerea unor repere cu toleranțe foarte mici, cum ar fi, de exemplu, piesele pentru ceasuri, a fost pînă nu demult determinată de operații de mare precizie, executate de specialiști de cea mai înaltă clasă. În prezent, această imagine este în schimbare, datorită în principal folosirii controlului numeric, controlului numeric cu ajutorul calculatorului, controlului numeric direct, precum și a roboților în prelucrările cu mașini-unelte.

Obiectivele introducerii mijloacelor de automatizare și robotizare au fost legate de eliminarea „timpilor morți", reducerea costurilor de producție — directe și indirecte —, utilizarea la maximum a utilajelor și creșterea generală a productivității. Liniile de transfer, dezvoltate în acest scop, a căror funcție constă în deplasarea produsului în conformitate cu etapele fluxului tehnologic, în trepte cu o viteză bine calculată, au unele dezavantaje cum ar fi limitarea vitezei de cea mai lungă operație sau specializarea strictă, ceea ce determină uzura morală rapidă. Ca urmare, au început să fie introduși roboții industriali, în principal pentru încărcarea și descărcarea mașinilor-unelte, rezultînd reducerea forței de muncă necesare.

Modelul de lucru cel mai utilizat în aplicații cu mașini-unelte este reprezentat de așezarea roboților — în centru, înconjurați de utilajele pe care le deservesc — în număr de cel puțin trei. De altfel dacă timpul de execuție a unei operații depășește o anumită valoare (10—20 secunde), neocuparea robotului cu sarcini suplimentare conduce la pierderi economice.



În prezent au apărut roboți capabili să efectueze deplasări între mașinile deservite, ceea ce permite creșterea numărului acestora ; pentru eliminarea limitării cauzate de timpul diferit de lucru al fiecărui utilaj sînt prevăzute stații tampon cu piese suplimentare. De menționat că roboții mobili asigură posibilitatea folosirii unor spații mai reduse de lucru decît cei staționari deoarece, de obicei, mașinile-unelte din liniile de transfer sînt așezate una lîngă alta și accesul este mai simplu.

Pentru creșterea productivității se utilizează în mod frecvent roboți cu instalații duble de apucare și plasare a pieselor.

Liniile tehnologice prevăzute cu roboți au o productivitate mai mare cu cel puțin 25% față de operațiile manuale, recuperarea cheltuielilor de investiții făcîndu-se în circa 2 ani.

Este mai puțin cunoscut faptul că, în Japonia, țara cu cei mai mulți roboți instalați, ponderea o reprezintă aplicațiile cu mașini-unelte, estimîndu-se că în curînd vor apare fabrici complet robotizate în care produsul final va putea fi cunoscut după programele ce se vor introduce la un moment dat în calculatoarele și roboții din liniile de producție.

Roboții folosiți în domeniul prelucrărilor cu mașini-unelte prezintă unele cerințe suplimentare, cum ar fi un număr de pînă la șase articulații, posibilitatea reprogramării pentru preluarea rapidă a pieselor din diferite locuri, precizia repetabilității mișcării, viteza mare de lucru, selecția programelor în funcție de mașinile sau stațiile de control deservite, compatibilitatea cu sisteme complexe *NC*, *DNC*, *CNC*, posibilitatea paletizării și depaletizării, acceptarea atît a programelor de casă cît și a celor din biblioteci, fiabilitatea mărită (MTBF de peste 400 ore), posibilitatea montării instalațiilor duble de apucare și plasare a pieselor, mobilitatea în spațiul de lucru, sesizarea automată a poziției pieselor asimetrice, adaptabilitatea la mediu, inclusiv recunoașterea imaginilor etc.

Robotica și comenzile numerice pot fi considerate ca fiind complementare în tehnologiile de control, ambele conducînd la tehnologii de fabricație mai complexe care vor sta la baza producției complet automatizate.

### 4.5.6. PALETIZAREA ȘI DEPALETIZAREA

În numeroase cazuri produsele și subansamblele rezultate după o anumită operație sînt paletizate, pentru a fi transportate mai ușor la alte locuri de muncă sau în magazie ; de asemenea, deservirea mașinilor cu astfel de produse și subansamble se face prin depaletizare. Se menționează că, pe de o parte, obiectele manipulate sînt de dimensiuni, forme și greutăți diferite iar, pe de altă parte, la cererea beneficiarilor sînt paletizate produse diferite de catalog, într-un singur colet.

Roboții au demonstrat deja calități deosebite pentru realizarea cu mare viteză și precizie a paletizării și depaletizării, mai ales în cazurile de schimbare a modelului de așezare, cînd este necesară o simplă selectare de programe.

La paletizare, atunci cînd robotul lucrează în strînsă dependență cu un sistem de control, pot să apară următoarele situații :

- încărcarea completă a unei palete goale ;
- încărcarea limitată a unei palete goale ;
- completarea unei palete parțial încărcate ;
- adăugarea unui număr limitat de obiecte pe o paletă incomplet încărcată.

Din cele de mai sus rezultă atît complexitatea operației ce urmează a fi efectuată de robot, cît și necesitatea ținerii unei evidențe exacte a numărului de produse paletizate. Uneori programele roboților trebuie să țină seama și de alte considerente, cum ar fi, de exemplu, posibilitatea schimbării modelului de așezare între două straturi succesive, pentru a se asigura o mai mare stabilitate a paletei.

Problema care se pune în prezent este ca roboții să recunoască situațiile cu care se confruntă pentru a selecta programul cel mai potrivit ; astfel de posibilități



necesită senzori tactili sau de recunoaștere vizuală — elemente care fac obiectul generațiilor noi de echipamente.

Depaletizarea este, de asemenea, un proces în care au fost introduși roboții, cu toate că uneori pot să apară situații deosebite, cum ar fi cazurile în care paleta este incompletă sau numărul de obiecte de pe o paletă completă este insuficient pentru realizarea operației, mai ales cînd urmează a fi obținute anumite amestecuri foarte bine dozate în chimie sau în alte domenii. Rezolvarea acestor sarcini se poate face prin programarea *off-line* a robotului, combinată cu prevederea unui panou special pentru operator în vederea luării deciziei optime într-un mod cît mai rapid.

Complexitatea unor paletizări și depaletizări, precum și problemele legate de greutatea, volumul și forma obiectelor manipulate fac necesară de multe ori utilizarea unor roboți cu o structură tot mai complexă.

### 4.5.7. APLICAȚII NUCLEARE

În domeniul nuclear roboții sînt priviți cu un deosebit interes datorită, în principal, posibilităților mari pe care le oferă pentru înlocuirea, în condițiile grele de exploatare, a forței de muncă umane și pentru îmbunătățirea indicatorilor economici. Operațiile care pot fi robotizate cu prioritate constau în supravegherea, întreținerea și efectuarea reparațiilor centralelor nuclearo-electrice. Cu roboții existenți s-a reușit să se intervină în activitatea de supraveghere a proceselor nucleare, dar nu s-au găsit încă soluțiile cele mai potrivite pentru efectuarea întreținerilor și reparațiilor, decalajul față de primul caz fiind de circa trei ani.

În centralele nuclearo-electrice și-au găsit deja locul roboți specializați pentru tăierea țevilor, sudura, inspecția și repararea tuburilor de la generatorul de aburi, precum și pentru inspecția ultrasonică. Dacă se au în vedere posibilitățile de comercializare, atunci cele mai avansate sînt sistemele de inspecție. Principalul argument privind utilizarea acestora în domeniul nuclear

îl reprezintă potențialul mare de acces în condiții de radiații, temperaturi și presiuni înalte, nepotrivite pentru activitatea oamenilor. Roboții permit scurtarea timpului de întrerupere a funcționării pentru intervenții fie prin efectuarea unor operații care elimină pregătirile necesare în cazul accesului factorului uman, fie prin organizarea locului de muncă și facilitarea în acest fel a activităților în zonele periculoase. În prezent multe din dificultățile implicate de utilizarea roboților în centralele nucleare provin din faptul că, anterior, la punerea lor în funcțiune nu s-au avut în vedere marile posibilități oferite de robotică. În viitor obiectivele privind dezvoltarea acestui domeniu vor trebui să fie astfel proiectate încît să permită efectuarea de către roboți a operațiilor de supraveghere și întreținere prin includerea de spații potrivite de acces și intervenție, posibilități de iluminare și alimentare cu energie electrică, precum și prin montarea optimă a echipamentelor și instalațiilor astfel încit acestea să nu constituie obstacole pentru activitatea roboților în caz de necesitate.

Studiile întreprinse au arătat că roboții pot executa mai mult de 20 de operații diferite, de la realimentarea cu combustibil, pînă la întreținerea barelor de control, repararea tuburilor de la generatorul de aburi și înlocuirea sau remedierea defecțiunilor la pompe sau supape. În cele ce urmează se prezintă principalele tipuri de roboți cu aplicații în domeniul nuclear.

• Începînd din anul 1979 în centralele nucleare se utilizează pe larg manipulatoare Vermaat, produse în Olanda, pentru inspecția și repararea tuburilor de la generatorul de aburi ; inițial acestea au fost proiectate numai pentru testarea și examinarea vizuală a tuburilor și pentru măsurarea profilelor.

Dispozitivele respective sînt instalate la capătul canalului de la generatorul cu aburi, între pereții acestuia și tubulatură, cu ajutorul a trei elemente de fixare. Se pot efectua operații de identificare și marcare a tuburilor, reparații, tratamente termice, curățire chimică, sudură etc. Sistemul de control cu microprocesor permite programarea lucrului la un număr de 100 de tuburi.



Principalele avantaje oferite de cele peste 20 tipuri de manipulatoare Vermaat, în special pentru generatoare de tip Westinghouse, dar și pentru alte tipuri, ca de exemplu Candu, sînt instalarea și demontarea rapidă, fără a se pătrunde în generatorul de aburi, poziționarea precisă și într-un termen scurt — cu acces imediat la toate tuburile —, reducerea drastică a dozei de radiații la care este expus personalul etc.

• La centrala nucleară de la Three Mile Island, roboții au început să fie utilizați după accidentul care s-a produs în anul 1982. Robotul Sisi (*S*ystem *I*n- *S*ervice *I*nspection), în greutate de 11 kg, a fotografiat și a culeș date asupra iradierii din zonele apropiate, Fred (181 kg) a participat la decontaminarea pereților într-o cameră de pompe din clădirea auxiliară, iar Louie — construit de firma Westinghouse încă din anii '50 ,avînd trei camere de televiziune și fiind controlat de la distanță — a fost adus la locul accidentului pentru operații de curățire.

Robotul cel mai avansat care lucrează la centrala atomică menționată poate fi considerat Rover sau RRV (*R*emote *R*econnaissance *V*ehicle — vehicul de recunoaștere de la distanță). Rover — în greutate de 450 kg și posedînd 6 roți — a fost realizat prin cooperare de către cinci firme, fiind destinat pentru examinarea radiologică a subsolului reactorului. Controlul se face de la o distanță de peste 150 m de către două persoane. Alte tipuri de roboți, cum ar fi Moose și Rosa, sînt prevăzute a fi utilizate în reactoarele de la Three Mile Island în viitorul apropiat.

• Robotul Rosa (*R*emotely *O*perated *S*ervice *A*rm — braț pentru service comandat de la distanță) a început să fie utilizat de firma Westinghouse, începînd din septembrie 1983, în generatoare de aburi din reactoarele cu apă sub presiune, putînd lucra în imersiune și în zone cu radiații înalte de peste 1 000 rad/h. Brațul cîntărește circa 5 kg, are 6 segmente, tot atîtea grade de libertate și ridică peste 23 kg la o extensie de 2,1 m.

Un robot Rosa este alcătuit din 4 subsisteme : brațul de operare telecomandat, calculatorul de control inclusiv monitorul video, unitatea suport și subsistemul

de execuție ; fiecare dintre aceste elemente este alcătuit la rîndul său din diverse module specifice aplicațiilor.

• Inspecția și repararea generatoarelor de aburi se poate face cu manipulatoare din generația a treia proiectate de firma Babcock & Wilcox. Acești roboți sînt cunoscuți sub numele de Roger (*R*emotely *O*perated *G*enerator *Ex*amination and *R*epair) și pot efectua în plus laminarea capetelor tuburilor, sudura de la distanță, detectarea scurgerilor radioactive etc. Manipulatorul a fost prezentat în octombrie 1983 și s-a bucurat de succes datorită simplității instalării, fiind utilizat în numeroase centrale nucleare. Sistemul de control cu microprocesor și comanda motorului pot fi plasate la distanță de robot ; sarcina utilă este de peste 90 kg.

• Sistemul de manipulare și control la înlocuirea tuburilor în centralele nucleare de tip Candu necesită echipamente pentru poziționarea sculelor, transferul componentelor, precum și controlul și monitorizarea procesului în ansamblu, fiind constituit din următoarele subsisteme : stația de lucru de la distanță, telemanipulatorul, subsistemul de observare a bolții, subsistemul de transmisie a datelor, centrul de teleoperare.

Firmele Atomic Energy of Canada Ltd., Canadian General Electric, Spar și Ontaria Hydro proiectează și livrează echipamentele pentru procesele de înlocuire manuală sau telecomandată a componentelor din tuburile de combustibil ale centralelor CANDU-PICKERING „A".

Mecanismele de reutilare sînt montate prin intermediul unei interfețe universale pe un cadru mobil — stație de lucru care are rolul de aliniere a sculelor în dreptul canalului. În general se montează patru mecanisme care pot fi schimbate în mod independent. Stația de lucru asigură două grade de libertate de translație și trei de rotație, precum și indexarea capului revolver. Poziția inițială este fixată cu o sculă de aliniere, datele respective fiind memorate pentru ca toate sculele să poată fi poziționate cu precizie. Numai o singură sculă de pe o stație de lucru este operațională la un moment dat.

Sistemul funcțional este constituit din cîte două stații de lucru montate pe fiecare din cele două părți ale reactorului, cu posibilitatea de a fi poziționate paralel



cu acestea prin mișcări înainte-înapoi în interiorul unui cabinet de izolație. Cabinetul are o structură de oțel inoxidabil și este montat pe podul mașinii de alimentare pentru a permite accesul operatorilor, în costume izolate, la fața reactorului în vederea efectuării unor operații manuale. De asemenea cabinetul prezintă unele ferestre care se deschid în dreptul canalelor și macaralelor pentru schimbarea capetelor revolver pe stația de lucru.

Transferul componentelor de la sculele de pe podul mașinii de alimentare la transportoarele dispuse pe sol și invers se face cu ajutorul unui braț cu mai multe încheieturi, controlate prin servomecanisme de un sistem electrohidraulic comandat cu ajutorul unui calculator electronic. Întregul ansamblu este montat pe un cărucior care se poate mișca pe șine dispuse în paralel cu partea din față a reactorului. Prin comenzi simple se pot controla viteza și direcția deplasării unor sarcini de peste 1,3 tone.

Sistemul de control se află în centrul de telecomandă situat la o distanță de pînă la 500 m. El este acționat de șase operatori — cîte unul pentru cele patru stații de lucru, un manipulator și un supervizor. În cazul în care stațiile de telecontrol nu sînt operaționale, fiecare subsistem poate fi controlat prin cîte o unitate locală portabilă.

Accesul audio-vizual pe tot parcursul operației este asigurat prin microfoanele și camerele de luat vederi ale unui sistem de televiziune în circuit închis de înaltă rezoluție.

Operația de retubare este controlată de un set specific de instrucțiuni și opțiuni de control. Setul de instrucțiuni cere operatorului să realizeze anumite acțiuni, validează datele de la operator, verifică executarea fiecărei instrucțiuni și permite efectuarea comenzilor ce provin de la mai mulți operatori, putîndu-se asigura chiar automatizarea diverselor secvențe.

• Firma Remote Technology Corp. (Remotec) și-a orientat eforturile în direcția realizării de roboți pentru supraveghere și inspecție. Roboții de acest tip sînt alcătuiți din senzori, un sistem de prindere, un transpor-

tor și consola de control. Unele sarcini actuale ale personalului, cum sînt monitorizarea și supravegherea de rutină pot fi îndeplinite de echipamentele și senzorii existenți, prin reproiectare. În acest fel s-ar reduce atît pericolul expunerii la radiații a personalului cît și costurile de operare. Este posibilă selectarea senzorilor care să înlocuiască unele activități ale omului — supraveghere vizuală, măsurarea nivelelor de radiație etc. —, dar apar probleme în cazul transportării și amplasării senzorilor în diferite arii cu radiații ; se menționează că se obțin costuri reduse pentru roboții care pot fi utilizați în mai multe arii de lucru. În urma analizei efectuate, Remotec a trecut la proiectarea și fabricarea unor roboți specifici, adaptați condițiilor din diferite centrale. Un asemenea robot, denumit Surbot *(surveillance robot)*, este utilizat la centrala nucleară TVA-Brown Ferry.

● Pentru medii care prezintă riscuri firma Advanced Resource Development Corp. (ARD) a realizat sistemul de inspecție industrială de la distanță, IRIS *(Industrial Remote Inspection System)*, avînd dimensiuni relativ reduse (lungimea egală cu 107 cm, lățimea cu 56 cm iar înălțimea de 36 cm). IRIS este de fapt un transportor alimentat de la baterii, pe care se pot monta senzorii optici, audio și de mediu, manipulatoarele, precum și subsistemele de comunicații de date și control. Sistemul mai cuprinde un braț telescopic, un sistem TV tridimensional avînd lentile zoom și microfoane. Sarcina utilă a robotului este de 32 kg.

Sistemul IRIS integrează și o anumită inteligență pentru a se putea retrage dacă este necesar, chiar dacă semnalele manuale de control se pierd sau sînt blocate prin interferențe. Testele efectuate au demonstrat posibilități bune de utilizare în diferite tipuri de centrale nucleare.

● Cercetări importante privind sistemele de inspecție se desfășoară în Japonia, urmărindu-se, în principiu, realizarea unor dispozitive care să asigure mobilitatea roboților. Firmele Toshiba, Mitsubishi și Hitachi sînt printre cele mai avansate în acest domeniu. Robotul LCR-1 *(Ladder Climbing Robot)*, realizat de Toshiba, se



poate mișca în sus sau în jos cu o viteză de 4 m/min. și poate duce greutăți de circa 7 kg, la o greutate proprie de 32 kg și dimensiuni de 1 m în lungime, 35 cm în lățime și 26 cm în înălțime. Acest robot poate fi utilizat în locuri inaccesibile pentru alte tipuri, în vederea inspectării conductelor, cablurilor etc.

Firma Mitsubishi a proiectat un robot care combină picioarele și roțile pentru a putea pătrunde în interiorul reactorului. Denumit C/V *(Containment Vehicle)*, robotul are patru picioare cu două încheieturi, două roți de tracțiune și patru roți suport la capetele picioarelor, precum și un manipulator și camere TV stereo. În funcție de mediu robotul poate merge pe roți sau pași peste obstacole.

● Reactoarele Chinon A3 aparținînd firmei Electricité de France utilizează manipulatoare pentru inspecție și reparații. Manipulatoarele au fost proiectate în cooperare de Hispano Suiza și cercetătorii de la centrala nucleară menționată. Lucrările au fost determinate de fenomenul de oxidare a componentelor din oțel cu procent redus de carbon în atmosfera de bioxid de carbon la înaltă presiune, folosit ca agent de răcire. Datorită dificultăților de acces și necesității reparării rapide prin întărirea structurilor sudate sau buloanelor s-a ajuns la soluția unui braț telescopic de 11 m, denumit ISIS, care poate trece printr-un canal de alimentare cu lungimea de 7 m și diametrul de 0,22 m, pentru a se ajunge la zona de deasupra inimii reactorului ; sarcina utilă este de 70 kg la 2,5 m de coloană și 150 kg la 1 m.

Brațul de acționare are trei umeri și trei încheieturi care asigură 8 grade de libertate ; în momentul acționării, două grade de libertate sînt fixate, iar șase sînt controlate simultan de calculatorul electronic. Se pot, astfel, pregăti suprafețele pentru sudură, existînd posibilitatea de a executa suduri, găuri, ridicarea și poziționarea componentelor. Capetele telemetru de tip video folosesc și laserul pentru simulare, realizîndu-se precizii de poziționare de 0,3 mm.

Robotul are trei moduri de operare : telecomandă — cu sistem de control „master", utilizat pentru fixarea celor șase grade de libertate —, un mod automat, în care

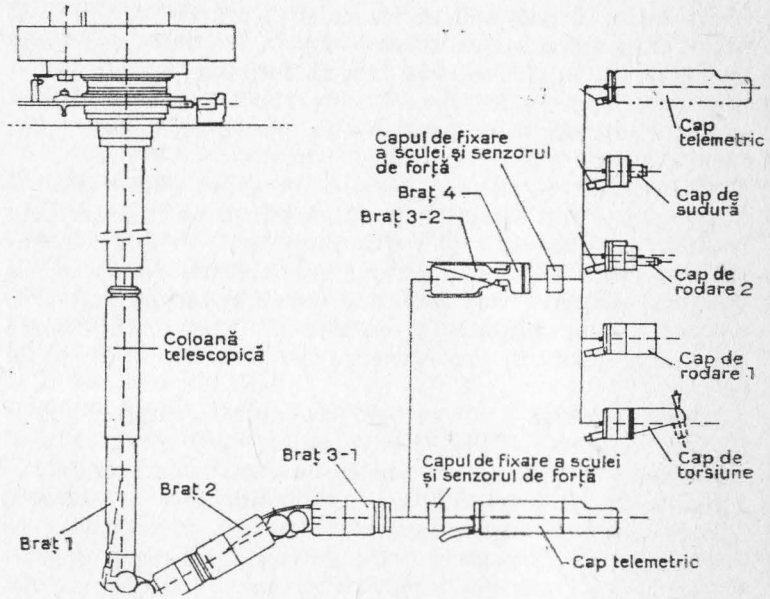



*Fig. 4.9.* Robotul ISIS.

manipulatorul reproduce mișcările învățate, colectînd sculele și piesele de lucru și un al treilea mod, în care manipulatorul poziționează final piesa de lucru în reactor.

În fig. 4.9 se prezintă schematic robotul ISIS.

Din cei cinci roboți ISIS utilizați în centrala Chinon, doi sînt folosiți pentru simulare, iar trei în reactor.

• În luna martie 1983 a avut loc prezentarea demonstrativă a robotului numit Odex I, produs de firma Odetics Inc. Acest robot poate acționa în toate mediile, utilizînd roți, șine sau picioare ; el este controlat de un calculator electronic propriu și execută o mare varietate de funcții utile. Robotul „pășește" cu ajutorul a două seturi de cîte trei picioare ; un set avansează, în timp ce al doilea susține robotul.

Odex I poate depăși obstacole înalte de pînă la 84 cm și ridică greutăți de 945 kg în modul staționar și 450 kg



la viteza normală, greutatea proprie fiind de numai 166,5 kg. Alimentarea independentă cu energie și comunicațiile prin radio îi asigură o libertate totală de mișcare.

Rezultatele remarcabile obținute de Odex I au condus recent la numeroase comenzi adresate firmei producătoare pentru realizarea unor sisteme mai evoluate. Noul echipament proiectat, denumit „SRL Walking Robot", va avea în plus unele componente — manipulatorul, sistemul TVCI, o consolă pentru telecontrol și un sistem electric de alimentare. Manipulatorul constă dintr-un braț flexibil montat pe platforma superioară rotativă ; la încheietură sînt plasate un gripper și o cameră TV cu sistem de iluminare. Manipulatorul acționează pe toată circumferința la o înălțime de peste 2 m de la sol. Camera TV asigură informații pentru operatorul de la consolă printr-un monitor.

Robotul descris mai sus reprezintă un pas important în direcția utilizării sistemelor complexe în centralele nucleare.

Sînt necesare, totuși, o serie de îmbunătățiri : televiziunea stereo, tehnicile de măsură în infraroșu, acustică și laser pentru asigurarea unor informații adecvate, în mod deosebit în medii cu vizibilitate redusă. Avînd în vedere posibilitatea de a lucra în medii dintre cele mai ostile, multe din componentele sistemului descris trebuie să fie ecranate la efectele radiațiilor. De asemenea trebuie prevenită contaminarea internă și asigurată decontaminarea rapidă a exteriorului.

Avantajele roboților de tip Odex față de alte variante sînt cu atît mai însemnate cu cît, în reactoarele cu apă, sînt necesare unele operații subacvatice referitoare la realimentare, precum și la depozitarea și manevrarea combustibilului uzat. Totodată inspecția și întreținerea conductelor fac necesară combinarea mersului cu suspendarea și rostogolirea atît în interiorul, cît și în exteriorul acestora.

• Firma Toshiba și Universitatea din Tokyo au realizat în colaborare un prototip de robot mobil inteligent, capabil să urce scările ; denumit „Amooty" — de la ini-

țialele cercetătorilor — robotul este utilizat pentru inspecția și întreținerea centralelor nuclearo-electrice.

În interiorul centralei, Amooty se poate mișca automat pe baza unei hărți memorate anterior ; robotul utilizează senzori pentru verificarea poziției sau direcției mișcării. Cînd apar obstacole robotul le identifică și le depășește ori le înconjoară pentru a-și îndeplini misiunea. Pe scurt, Amooty posedă echivalentul picioarelor, mîinilor și ochilor omului, la care se adaugă un sistem de control integrat utilizînd un limbaj dezvoltat recent, denumit *Amorl (Amooty Robot Language)*.

Mecanismul de transport (fig. 4.10*a*) este constituit din patru „picioare" — sisteme de cîte trei roți care se mișcă independent. Cele două roți de la nivelul solului asigură mișcarea prin rotație. Pentru a urca sau coborî scările ori obstacole de pînă la 22 cm brațele radiale se rotesc (fig. 4.10*b*) ; viteza de cățărare este de 37,5 s/pas la înclinări de 35°, iar la sol atinge 1 km/oră. Baza roților poate fi modificată în limitele a 160 cm la urcarea scărilor și cu pînă la 80 cm la întoarcerea într-un spațiu redus (fig. 4.10*c*).

Manipulatorul are șase grade de libertate de bază plus trei grade suplimentare, asigurîndu-se astfel o gamă largă de mișcări. Robotul este echipat cu o cameră TV și un dispozitiv laser, montate pe o masă rotativă, pentru măsurarea distanțelor. Poziția poate fi calculată prin recunoașterea unor simboluri cu reflexie scăzută dispuse de-a lungul traiectoriei.

Robotul Amooty dispune de un sistem de control integrat și subsisteme de comandă cu microprocesoare Z80, de un manipulator cu microprocesoare 8086 și senzori vizuali cu LSI-11/23. Subsistemele individuale pot fi mișcate independent sau corelat cu ajutorul microcalculatorului. Sistemul de control este automat, fiind asigurat de calculatorul electronic cu ajutorul limbajului Amorl, sau manual prin intermediul unui joystick.

Studiile întreprinse referitoare la avantajele folosirii roboților pentru curățarea cavităților reactoarelor (roboți submersibili care lucrează pentru îndepărtarea impurităților de pe pereți), examinarea condițiilor de mediu (roboți de forma unor platforme mobile cu posi-



bilitatea deplasării pe scări și peste obstacole, pe care sînt fixate echipamente de inspecție și calculatoarele de prelucrare a datelor), strîngerea/desfacerea șuruburilor la flanșe (dispozitivul fiind transportat de o platformă robot) și sudura de la distanță au scos în evidență amortizarea după 1—3 ani a costurilor de achiziționare.

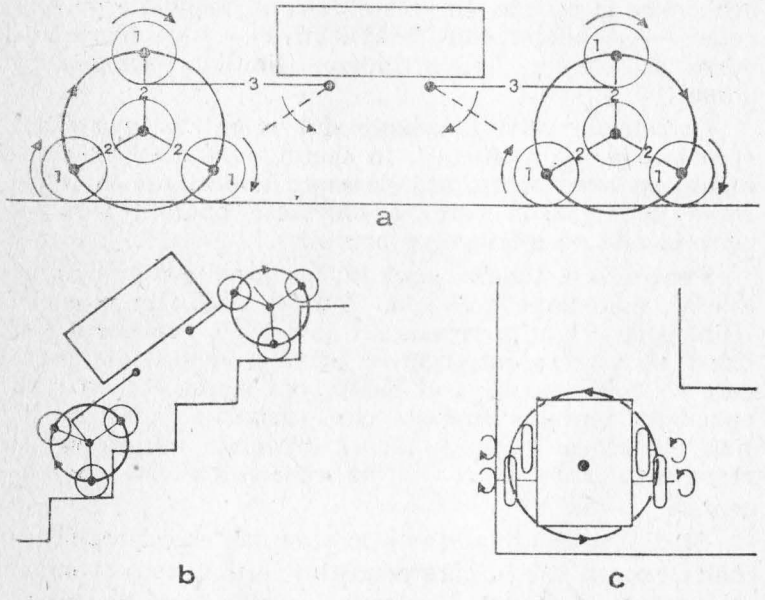



*Fig. 4.10.* Robotul Amooty :
*a* — deplasarea pe orizontală ; *b* — urcarea scărilor ; *c* — întoarcerea.

Aplicațiile din domeniul nuclear sînt mult mai potrivite pentru roboții cu picioare decît pentru cei cu roți sau șenile ; cercetările actuale întreprinse în U.R.S.S., S.U.A. și Japonia se concentrează asupra sistemelor cu picioare, dar realizarea lor în număr mare rămîne o problemă de viitor. Caracteristicile mecanice, în principal manipulatoarele și brațele roboților, precum și raportul putere/greutate, reprezintă de asemenea probleme cărora viitorul le va da cu siguranță rezolvarea corespunzătoare.

## 4.6. ROBOȚII ȘI SISTEMELE DE INTELIGENȚĂ ARTIFICIALĂ

Robotica, împreună cu tehnologia informației și biotehnologia reprezintă discipline noi care au un potențial ridicat și influențe majore asupra dezvoltării activităților economice și sociale. În aceste domenii, lucrările de cercetare — fundamentale și aplicative — sînt foarte diverse și conduc la extinderea limitelor cunoașterii umane.

Cercetările privind roboții sînt orientate în prezent spre inteligența artificială, în speranța că acest nou domeniu va avea contribuții de mare importanță la realizarea unor mașini care să efectueze operații similare celor bazate pe inteligența omului.

Pentru investigarea gradului de „inteligență" al unei mașini, matematicianul Alan Turing, unul din pionierii tehnicii de calcul, a propus, în anul 1950, următorul test bazat pe tehnica interogării : printr-o modalitate oarecare se comunică separat cu un om și respectiv cu un calculator, fără a se cunoaște cine răspunde la întrebările puse ; dacă nu poate fi făcută distincția asupra sursei răspunsurilor se poate afirma că mașina este „inteligentă".

Stadiul actual de dezvoltare a permis abordarea sistemelor expert, caz în care cunoștințele unui expert uman sînt reduse la un set de reguli. Un calculator electronic adecvat poate utiliza aceste reguli, constituite într-o bază de cunoștințe, prin intermediul unei mașini de inferențe, rezultînd concluzii logice corespunzătoare. Sistemele-expert intră în aria de preocupări majore legate de proiectele privind calculatoarele din generația a cincea și vor deveni cu siguranță unelte importante în proiectarea roboților avansați. La realizarea acestui deziderat o contribuție însemnată o vor avea sistemele de software. În fig. 4.11 sînt ilustrate principalele forme de software și locul ocupat de sistemele de inteligență artificială în domeniul roboților.



Un loc important în acest ansamblu îl vor avea senzorii care sînt în curs de perfecționare și integrare de funcții noi.

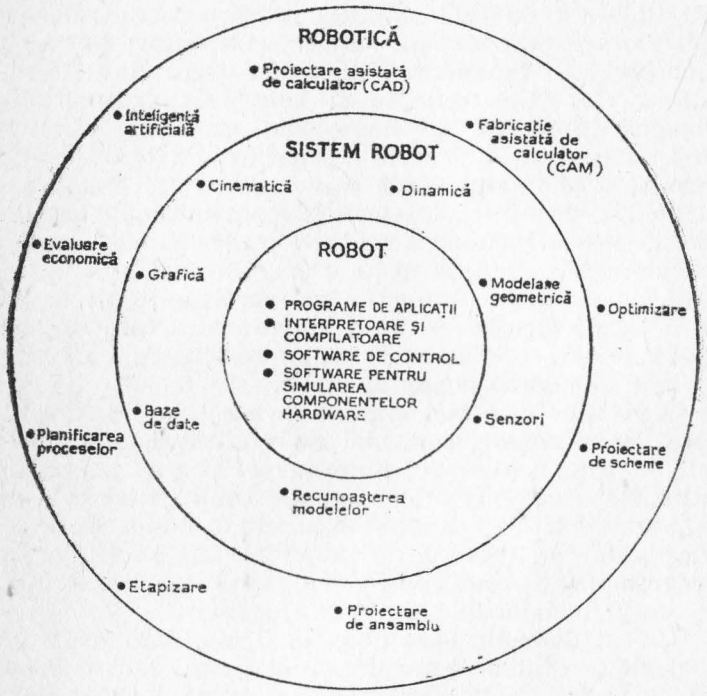



*Fig. 4.11.* Clasificarea formelor de software în robotică.

Senzorii inteligenți vor combina practic în mod unitar avantajele tuturor tehnologiilor utilizate separat pînă în prezent.

De la sistemele compuse din senzor, formator de semnal, convertor A/D și microprocesor s-a trecut la integrări de tipul senzor-formator de semnal, convertor A/D-micropocesor și senzor-formator de semnal-convertor A/D, tinzîndu-se spre senzorii inteligenți care urmează să integreze toate cele patru componente. Noul tip de senzor va avea multe caracteristici care nu sînt posibile

la variantele convenționale, datorită în principal funcțiilor de prelucrare complexă a semnalului, memoriei asociate și tehnicii de calcul înglobate.

Pentru roboți sînt în curs de realizare senzori tactili liniari, bidimensionali de înaltă rezoluție, utilizînd calculatoare paralele, precum și senzori inteligenți care încorporează funcțiunile unităților centrale ale calculatoarelor. Un exemplu îl constituie procesorul de imagine inteligent, tridimensional, pe o singură structură, preconizat în cadrul proiectelor MITI (Japonia), prin care se intenționează să se reproducă simțul tactil al omului cu ajutorul tehnologiilor semiconductoare. În viitor, senzorii cu semiconductori vor fi combinați cu calculatoarele electronice pe o singură structură în aplicații ale circuitelor integrate pe scară foarte largă. Senzorii tactili și senzorii pe o singură structură condiționează practic realizarea roboților inteligenți care să se apropie de comportamentul uman.

Aplicațiile cele mai importante ale roboților fiind legate de asamblare, cele mai mari eforturi sînt îndreptate în această direcție, în principal pentru asamblarea automată a componentelor simple. Unul dintre cele mai avansate proiecte, destinat asamblării motoarelor electrice ,este cunoscut sub denumirea APAS (*A*daptable *P*rogrammable *A*ssembly *S*ystem) și se încadrează în categoria sistemelor de asamblare flexibilă.

Un alt domeniu care necesită flexibilitate este cel al vehiculelor ghidate automat, pentru care s-au dezvoltat roboți mobili, avînd posibilitatea folosirii unui sistem de navigație intern, precum și a unor hărți pentru păstrarea poziției și recunoașterea traseului de urmat. Realizarea unor astfel de instalații a determinat cercetări asupra senzorilor și tehnicilor de navigație, control și comunicații, multe dintre acestea bazîndu-se pe noi structuri software. În această categorie se încadrează și preocupările legate de proiectarea unor roboți la care sistemul de mișcare să fie constituit din picioare.

Fabricile automate reprezintă o etapă superioară în evoluția producției fără oameni. Dintre realizările mai deosebite se amintesc fabrica automată Fujitsu Fanuc pentru asamblarea motoarelor și servomotoarelor, precum și proiectul experimental al Universității Purdue denumit CIDMAC (Computer *I*ntegrated *D*esign *M*anufacturing and *A*utomation Centre).



Avantajele fabricilor automate, concepute pe baza roboților și a sistemelor de inteligență artificială, sînt evidente. Pentru atingerea acestui obiectiv sînt investite forțe umane și materiale uriașe, rezultatele făcîndu-și deja apariția.

După reducerea costurilor în tehnica de calcul și creșterea posibilităților dispozitivelor grafice de înaltă rezoluție și ale software-ului, a apărut posibilitatea utilizării calculatoarelor în proiectare în principal la pregătirea documentațiilor tehnice. Sistemele dezvoltate pentru proiectarea asistată de calculator (CAD) conduc la creșterea productivității de peste 3—4 ori.

Pe măsura creșterii fiabilității circuitelor microelectronice și microminiaturizării a devenit posibil transferul automat al datelor de inginerie la sistemele de fabricație asistată de calculator (CAM) obținîndu-se sisteme complexe CAD-CAM. Datorită introducerii masive a reacției negative bazate pe semnale de la senzori în domeniul roboților a apărut și posibilitatea suplimentară a inspectării și testării automate a produselor, realizîndu-se sisteme de proiectare, fabricație și testare asistate de calculator, CADMAT (Computer *A*ided *D*esign *M*anufacture *A*nd *T*esting).

Dezvoltările cele mai importante s-au produs în robotică, proiectare și fabricație asistate de calculator, precum și în domeniul sistemelor de fabricație flexibilă.

Sistemele de fabricație flexibilă, FMS (*F*lexible *M*anufacturing *S*ystems) reprezintă pasul următor al aplicațiilor cu roboți, mașini cu comandă numerică și calculatoare electronice. FMS se constituie într-un sistem de fabricație integrată care coordonează echipamentele menționate, combinat cu automatizarea, manipularea și monitorizarea materialelor. În acest fel, pe o singură linie de asamblare pot fi realizate mai multe loturi de produse diferite în condiții economice avantajoase. Spre deosebire de unele sisteme anterioare, FMS nu necesită linii sofisticate de asamblare, producția fiind descentralizată, iar piesele și subansamblele pentru montaj putînd fi

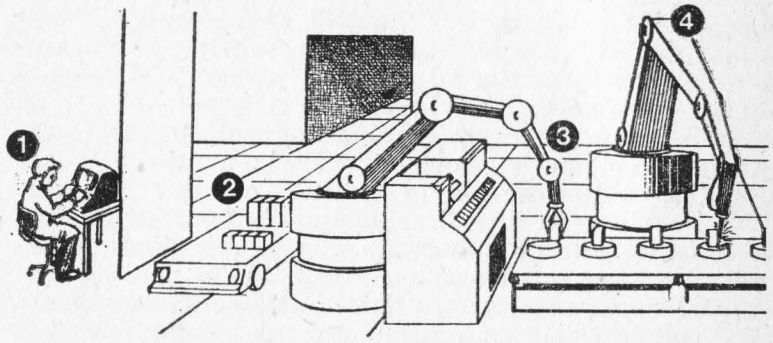
schimbate într-un timp foarte scurt (fig. 4.12). Un exemplu practic este ilustrat de Yamazaki Machine Works Ltd. care a conectat prin cabluri cu fibre optice modulul de comandă la 65 mașini-unelte asistate de calculator și 34 roboți, reducînd astfel de peste 10 ori forța de muncă necesară pentru producție.

Concepția globală — FMS și calculatoarele electronice — este cunoscută sub numele de fabricație integrată cu calculatorul, CIM (*Computer Integrated Manufacture*).

Calculatoarele au diverse sarcini în FMS și CIM, dar una dintre cele mai importante este reprezentată de posibilitatea asigurată pentru coordonatorii producției de a da comenzi și de a planifica sistemul pentru a permite prioritățile, astfel încît să se atingă cel mai înalt grad posibil de utilizare a mașinilor.

În viitorul apropiat se așteaptă să crească numărul domeniilor accesibile aplicațiilor cu roboți. Brațul de tip SCARA va deveni aproape comun, fiind dezvoltate în plus noi tipuri. În medie, roboții vor dispune de o putere de calcul apreciabilă, conținînd numeroase tehnici de programare textuală, inclusiv posibilități crescute de auto-instruire. Deși limbajul BASIC va fi generalizat datorită proliferării calculatoarelor personale, se vor utiliza pe larg limbajele avansate ; dintre acestea VAL ar putea deveni limbajul standard în domeniu.

Pe măsură ce complexitatea roboților se mărește vor deveni necesare limbaje avansate, cum ar fi, de exemplu, AML. Ar putea să apară chiar programatoare pentru roboți, cu funcții noi, deci care să poată trece efectiv la programare automată.






Dezvoltările înregistrate în tehnica de calcul, inspirate din proiectele privind generația a cincea de calculatoare electronice, vor simplifica sarcinile de programare, ceea ce va permite accesul tot mai larg al personalului la activitățile specifice roboților.

Domeniul asamblării va beneficia în cea mai mare măsură de roboți dotați cu sisteme vizuale cu mai multe niveluri de gri dar și color, tridimensionale, utilizați în prezent doar în aplicații speciale. De asemenea, puterea de calcul va fi utilizată pentru rezolvarea problemelor de recunoașterea și sinteza vorbirii.

În ansamblu calculatoarele electronice vor permite integrarea rapidă a producției industriale, ca și automatizarea generală. Cu toate acestea, fabricile fără personal uman vor fi destul de rare, cel puțin pînă la înțelegerea deplină a efectelor de integrare a producției la nivelul managementului.

Pînă la sfîrșitul deceniului actual vor apare în laboratoarele de cercetare roboții inteligenți, care constituie cea de-a treia generație de roboți.

Deși unele componente pentru roboți vor fi mai puțin costisitoare, pe ansamblu prețurile se vor situa la același nivel, cu precizarea că tehnica de calcul încorporată, deci și funcțiile, va fi superioară.

Peste circa cinci ani roboții vor avea posibilitatea manipulării unor materiale flexibile, cum ar fi textilele

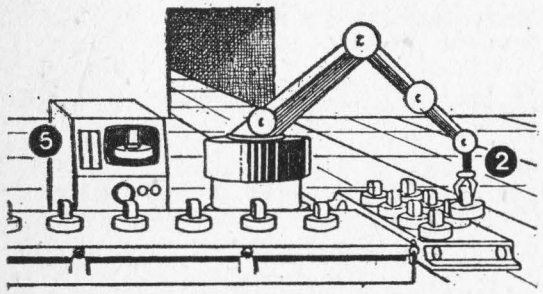



*Fig. 4.12.* Fabrica viitorului

*1* — Terminal de calculator ; *2* — Cărucior mobil ; *3* — Strung comandat de calculator ; *4* — Sudură ; *5* — Examinare în cameră cu raze x.

și cauciucul, caz în care software-ul necesar este destul de sofisticat. Roboții speciali — în principal sub forma unor dispozitive telecomandate manipulate de oameni și nu de calculatoare electronice — vor fi utilizați în medicină, dar și în mediul submarin sau în Cosmos, precum și în agricultură, mine sau centrale nucleare.

În aceeași perioadă vor fi introduse, cel puțin pentru experimentare, sisteme cu roboți automați, ghidați.

Se apreciază că în următorii 10—15 ani roboții vor pătrunde în domenii noi, ca de exemplu în activitățile casnice.

La sfîrșitul acestui secol vor apare adevărate sisteme integrate cu roboți sau echipamente complet automatizate. Se speră ca, în secolul următor, astfel de sisteme integrate să fie capabile de autoreproducere.

S-ar putea face numeroase aprecieri asupra viitorului sistemelor inteligente și al roboților. Reținem însă faptul deosebit de important că, în prezent, oamenii de știință pregătesc terenul pentru progresul semnificativ al inteligenței artificiale și roboticii, în cadrul unei adevărate noi revoluții industriale.

# Anexă

## EXPLICAREA TERMINOLOGIEI FOLOSITE ÎN DOMENIUL SITEMELOR DE INTELIGENȚĂ ARTIFICIALĂ

ALGORITM *(Algorithm)*: Procedură care garantează obținerea unei soluții corecte sau optime.

ATOM, în LISP (*Atom*): Simboluri, în LISP.

BAZĂ DE CUNOȘTINȚE (*Knowledge base*): Parte a sistemului bazat pe cunoștințe sau a sistemului-expert, care conține cunoștințele domeniului.

BAZĂ DE DATE *(Data base)*: Set de fapte, afirmații sau concluzii utilizate în corespondență cu partea IF a regulilor, într-un sistem bazat pe reguli.

CADRU (*Frame*): Metodă de reprezentare a cunoștințelor, care asociază caracteristici (atribute și valori) cu noduri reprezentînd concepte sau obiecte.

CALCULUL PREDICATELOR *(Predicate calculus)*: Limbaj formal al logicii clasice care utilizează funcții și predicate pentru a descrie relații între entități individuale.

CĂUTARE (*Search*): vezi EXAMINARE.

CUNOȘTINȚE (*Knowledge*): Informații constituite din fapte și reguli, care utilizate de un program de calculator conduc la un comportament inteligent.

CUNOȘTINȚE GENERALE (*General knowledge*): Cunoștințe privind modul de rezolvare a problemelor sau de interacțiune cu utilizatorul, constituite în mașina de inferențe.

CUNOȘTINȚE SPECIFICE *(Domain knowledge)*: Cunoștințe specifice unui anumit domeniu, reprezentînd elemente ale bazei de cunoștințe.

DEMON *(Demon)*: Procedură activată de modificarea sau accesul valorilor în baza de date.

EURISTICĂ *(Heuristics)*: Mod de lucru, prin reguli sau simplificări care limitează examinarea în vederea găsirii unei soluții acceptabile în domenii complexe.

EXAMINARE (*Search*) : Proces de urmărire specializată a unui set de soluții posibile la o anumită problemă pentru a găsi soluția optimă.

EXPERT (*Expert*) : Specialist care prin experiență este capabil să rezolve probleme complexe dintr-un anumit domeniu.

FACILITĂȚI DE EXPLICARE (*Explanation facility*) : Partea sistemului-expert care explică modul de găsire a soluțiilor și justifică pașii întreprinși în acest scop.

FACTOR DE SIGURANȚĂ (*Certainty factor*) : Număr prin care este exprimată încrederea în fapte sau reguli.

INGINERIA CUNOAȘTERII (*Knowledge engineering*) : Procesul realizării sistemelor-expert.

INTELIGENȚĂ ARTIFICIALĂ (*Artificial intelligence*) : Ramură a științei calculatoarelor care privește dezvoltarea programelor inteligente pentru calculatoare.

INTELIGENȚĂ NATURALĂ (*Natural intelligence*) : Facultate umană de a înțelege, de a cunoaște.

INTERPRETOR (*Interpreter*) : Parte a mașinii de inferențe care decide asupra modului de aplicare a cunoștințelor domeniului.

LIMBAJ FORMAL (*Formal language*) : Set de caractere de lungime dată format dintr-un vocabular finit de simboluri.

LIMBAJ NATURAL (*Natural language*) : Metodă standard pentru schimbul de informații între oameni.

LIMBAJ ORIENTAT PE PROBLEME (*Problem-oriented language*) : Limbaj de calculator destinat unei clase particulare de probleme.

LISTĂ (*List*) : Colecție de articole, închise în paranteze, fiecare articol fiind constituit dintr-un simbol sau altă listă.

MAȘINĂ DE INFERENȚE (*Inference engine*) : Colecție de cunoștințe generale asupra modului de rezolvare a unei anumite probleme sau de interacțiune cu utilizatorul.

METACUNOȘTINȚE (*Metaknowledge*) : Cunoștințe despre cunoștințe sau cunoștințe privind modul de lucru sau de realizare a raționamentelor de către sistemul-expert..

METODE BAZATE PE LOGICĂ (*Logic-based methods*) : Metode de programare care utilizează calculul predicatelor pentru a structura programul și a urmări execuția programului.



METODE BAZATE PE OBIECTE (*Object-oriented methods*) : Metode de programare bazate pe utilizarea unor articole numite obiecte, care comunică între ele prin mesaje sub forma unor transmisiuni globale.

METODE BAZATE PE PROCEDURI (*Procedure-oriented methods*) : Metode de programare care utilizează subrutine înglobate pentru organizarea și controlul execuției programului.

METODE DE ACCES (*Acces-oriented methods*) : Metode de programare care fac uz de examinări care generează noi calcule la schimbarea sau citirea datelor.

MIJLOACE DE REALIZARE A SISTEMELOR-EXPERT (*Expert System Building Tool*) : Complexul constituit din limbajul de programare și pachetul de programe de suport utilizat la realizarea sistemelor-expert.

PROGRAMARE AUTOMATĂ (*Automatic programming*) : Sisteme care pot scrie programe pentru calculatoare pornind de la descrieri.

PLANIFICATOR (*Scheduler*) : Parte a mașinii de inferențe care decide asupra ordinii de utilizare a cunoștințelor specifice.

RAȚIONAMENT SIMBOLIC (*Symbolic reasoning*) : Metode de rezolvare a problemelor bazate pe aplicarea strategiei și euristicii pentru manipularea simbolurilor care reprezintă concepte din lumea reală.

REGULĂ (*Rule*) : Metodă formală de specificare a recomandărilor, directivelor sau strategiei, exprimată prin *IF... THEN...*

REPREZENTARE (*Representation*) : Proces de formulare a unei probleme care să faciliteze găsirea soluției optime.

REȚEA SEMANTICĂ (*Semantic net*) : Metodă de reprezentare a cunoștințelor prin cadre, care constă într-o rețea de noduri (concepte sau obiecte) conectate prin arcuri care descriu relațiile dintre noduri.

SIMBOL (*Symbol*) : Șir de caractere care reprezintă anumite concepte din lumea reală.

SISTEM BAZAT PE CUNOȘTINȚE (*Knowledge-based system*) : Program în care cunoștințele sînt organizate separat în baze de cunoștințe și respectiv mașini de inferențe.

SISTEM-EXPERT (*Expert system*) : Program de calculator care utilizează cunoștințe ale unor experți pentru a prezenta performanțe de înalt nivel într-un domeniu restrîns.

SISTEME DE MANAGEMENT ALE BAZEI DE DATE *(Database management systems)*: Sisteme de calcul care permit stocarea și regăsirea informațiilor dintr-un anumit domeniu.

STRUCTURĂ ARBORE *(Tree structure)*: Mod de organizare a informației sub forma unui graf.

SPAȚIU DE EXAMINARE *(Search space)*: Setul tuturor soluțiilor posibile ale unei probleme.

# BIBLIOGRAFIE


1. BALTAC, V.; DAVIDOVICIU, A. ș.a., *Sisteme interactive și limbaje conversaționale. Utilizare și proiectare*, Editura tehnică, București, 1984.
2. DRĂGĂNESCU, M. și colectiv, *Inteligența artificială și robotica*, Editura Academiei Republicii Socialiste România, București, 1983.
3. TABARCEA, P.; GHIUR ,GH., *La frontierele electronicii*, Editura Militară, București, 1985.
4. FEIGENBAUM, A. EDUARD; McCORDUCK, PAMELA, *The Fifth Generation*, Addison-Wesley, Comp., 1983.
5. BARR, A., FEIGENBAUM, A. EDUARD; COHEN, R. PAUL, *The Handbook of Artificial Intelligence*, William Kaufmann, Inc., Los Altos, California ,1982.
6. McCORDUCK, PAMELA ,*Machines who think*, W. H. Freeman, San Francisco, California, 1979.
7. ILIESCU ,I., *A cincea generație. Note de lectură*, AMC nr. 44, Editura Tehnică, București, 1984.
8. SIEWIOREK ,P. DANIEL; BELL, C. GORDON; NEWELL, ALLEN, *Computer Structures and Examples*, McGraw-Hill Book company, Inc., 1982.
9. * * * *Inteligența artificială și robotica*, Editura Academiei Republicii Socialiste România, 1983.
10. * * * *Computer Design*, vol. 24, *3*, March, 1985.
11. * * * *Computer Design*, vol. 24, 7, July, 1985.
12. * * * *Computer Design*, vol. 24, *10* August 1985.
13. WATERMAN, A. DAN, *Guide to Expert Systems*, Addison Wesley Publishing Company, 1985.
14. ODOBLEJA, ȘTEFAN, *Introducere în logica rezonanței*, Scrisul Românesc, Craiova, 1984.
15. TURKLE, SHERRY, *The second self-computers and the human spirit*, Simon and Shuster, New York, 1984.

# CUPRINS


*INTRODUCERE* . . . . . 5

**1. INTELIGENȚA ARTIFICIALĂ** . . . . . 11

1.1. Conceptul de inteligență artificială. Inteligența artificială și inteligența naturală. Ingineria cunoașterii . . 11
1.2. Evoluția inteligenței artificiale . . . . . 15
1.3. Tehnici pentru rezolvarea problemelor . . . . . 28
1.4. Reprezentarea cunoștințelor . . . . . 40
1.5. Limbaje de programare . . . . . 45
1.5.1. Programe pentru realizarea sistemelor de inteligență artificială . . . . . 52
1.5.2. Limbaje bazate pe cadre . . . . . 58
1.5.3. Limbaje bazate pe logică . . . . . 62
1.5.4. Limbaje pentru reprezentări prin obiecte . . . 63
1.5.5. Limbaje bazate pe proceduri . . . . . 64
1.5.6. Limbaje bazate pe reguli . . . . . 66

**2. SISTEME DE CALCUL SPECIALIZATE** . . . . . 74

2.1. Structura sistemelor de calcul actuale . . . . . 76
2.1.1. Sistemul de calcul B 5000 . . . . . 80
2.1.2. Sistemul ILLIAC IV . . . . . 85
2.1.3. Sistemul CONTROL DATA 6600 . . . . . 90
2.1.4. Sistemul de calcul CRAY-1 . . . . . 93
2.1.5. Calculatorul IBM SYSTEM/360 . . . . . 95
2.2. Structuri VLSI . . . . . 97
2.3. Arhitecturi avansate de calculatoare . . . . . 101





2.3.1. Arhitecturile RISC — o alternativă a procesării paralele . . . . . 102
2.3.2. Procesarea de tip paralel . . . . . 104
2.3.3. Supercalculatoarele — sisteme care extind opțiunile procesării în paralel . . . . . 114
2.4. Tendințe în dezvoltarea sistemelor de calcul . . . 120

**3. SISTEME-EXPERT** . . . . . 126

3.1. Structura sistemelor-expert . . . . . 126
3.2. Baze de date și cunoștințe . . . . . 132
3.3. Aplicații ale sistemelor-expert . . . . . 135
3.3.1. Medicină . . . . . 141
3.3.2. Chimie . . . . . 147
3.3.3. Tehnică de calcul . . . . . 150
3.3.4. Electronică . . . . . 152
3.3.5. Inginerie . . . . . 156
3.3.6. Geologie . . . . . 157
3.3.7. Management . . . . . 159
3.3.8. Producție . . . . . 160
3.3.9. Agricultură . . . . . 161
3.4. Perspective de dezvoltare a sistemelor-expert . . . 161

**4. ROBOȚI** . . . . . 165

4.1. Scurt istoric . . . . . 167
4.2. Clasificarea și structura roboților . . . . . 172
4.3. Interfața om-mașină . . . . . 185
4.3.1. Recunoașterea vorbirii . . . . . 185
4.3.2. Sinteza vorbirii . . . . . 194
4.3.3. Prelucrarea limbajelor naturale . . . . . 196
4.3.4. Sisteme de vedere artificială . . . . . 200
4.4. Programarea roboților . . . . . 203
4.5. Aplicații ale roboților industriali . . . . . 217
4.5.1. Sudura prin puncte . . . . . 218
4.5.2. Turnarea . . . . . 220

4.5.3. Sudura cu arc . . . . . 221
4.5.4. Forjarea . . . . . 223
4.5.5. Deservirea mașinilor-unelte . . . . . 224
4.5.6. Paletizarea și depaletizarea . . . . . 226
4.5.7. Aplicații nucleare . . . . . 227
4.6. Roboții și sistemele de inteligență artificială . . . 238

*Anexă.* Explicarea terminologiei folosite în domeniul sistemelor de inteligență artificială . . . . . 245

*Bibliografie* . . . . . 249